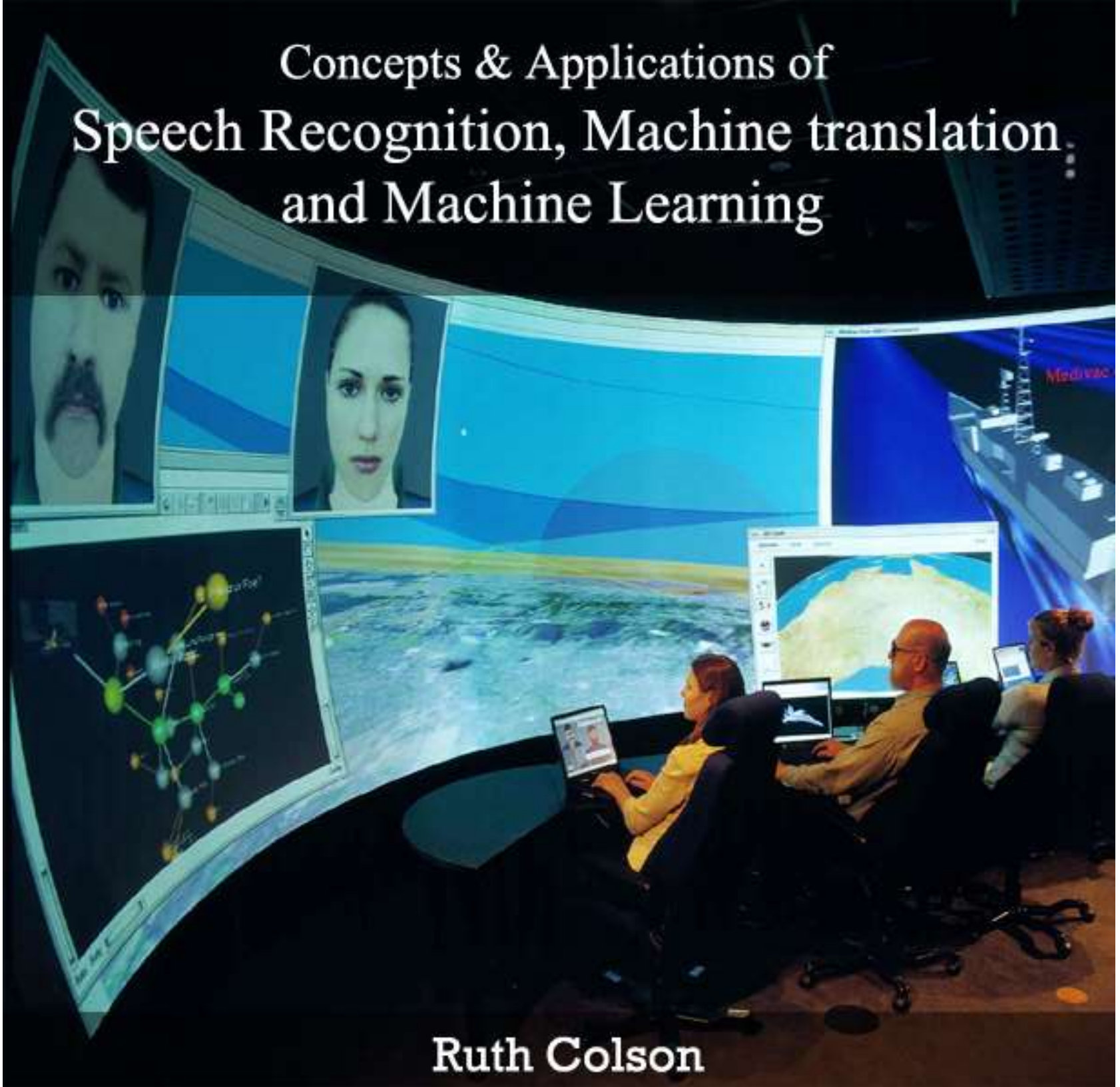
Concepts & Applications of Speech Recognition, Machine translation and Machine Learning



Ruth Colson

First Edition, 2012

ISBN 978-81-323-3527-6

Published by:
University Publications
4735/22 Prakashdeep Bldg,
Ansari Road, Darya Ganj,
Delhi - 110002
Email: info@wtbooks.com

Table of Contents

Chapter- 1

Speech Recognition

The display of the Speech Recognition screensaver on a laptop, in which the character responds to questions, e.g. "Where are you?" or statements, e.g. "Hello."

Speech recognition (also known as **automatic speech recognition** or **computer speech recognition**) converts spoken words to text. The term "voice recognition" is sometimes used to refer to recognition systems that must be trained to a particular speaker—as is the case for most desktop recognition software. Recognizing the speaker can simplify the task of translating speech.

Speech recognition is a broader solution which refers to technology that can recognize speech without being targeted at single speaker—such as a call center system that can recognize arbitrary voices.

Speech recognition applications include voice user interfaces such as voice dialing (e.g., "Call home"), call routing (e.g., "I would like to make a collect call"), domotic appliance control, search (e.g., find a podcast where particular words were spoken), simple data entry (e.g., entering a credit card number), preparation of structured documents (e.g., a radiology report), speech-to-text processing (e.g., word processors or emails), and aircraft (usually termed Direct Voice Input).

History

The first speech recognizer appeared in 1952 and consisted of a device for the recognition of single spoken digits Another early device was the IBM Shoebox, exhibited at the 1964 New York World's Fair. Lately there have been numerous improvements like a high speed mass transcription capabilitiy on a single system like Sonic Extractor

One of the most notable domains for the commercial application of speech recognition in the United States has been health care and in particular the work of the medical transcriptionist (MT). According to industry experts, at its inception, speech recognition (SR) was sold as a way to completely eliminate transcription rather than make the transcription process more efficient, hence it was not accepted. It was also the case that SR at that time was often technically deficient. Additionally, to be used effectively, it required changes to the ways physicians worked and documented clinical encounters, which many if not all were reluctant to do. The biggest limitation to speech recognition automating transcription, however, is seen as the software. The nature of narrative dictation is highly interpretive and often requires judgment that may be provided by a real human but not yet by an automated system. Another limitation has been the extensive amount of time required by the user and/or system provider to train the software.

A distinction in ASR is often made between "artificial syntax systems" which are usually domain-specific and "natural language processing" which is usually language-specific. Each of these types of application presents its own particular goals and challenges.

Applications

Health care

In the health care domain, even in the wake of improving speech recognition technologies, medical transcriptionists (MTs) have not yet become obsolete. The services provided may be redistributed rather than replaced.

Speech recognition can be implemented in front-end or back-end of the medical documentation process.

Front-End SR is where the provider dictates into a speech-recognition engine, the recognized words are displayed right after they are spoken, and the dictator is responsible for editing and signing off on the document. It never goes through an MT/editor.

Back-End SR or Deferred SR is where the provider dictates into a digital dictation system, and the voice is routed through a speech-recognition machine and the recognized draft document is routed along with the original voice file to the MT/editor, who edits the draft and finalizes the report. Deferred SR is being widely used in the industry currently.

Many Electronic Medical Records (EMR) applications can be more effective and may be performed more easily when deployed in conjunction with a speech-recognition engine. Searches, queries, and form filling may all be faster to perform by voice than by using a keyboard.

Military

High-performance fighter aircraft

Substantial efforts have been devoted in the last decade to the test and evaluation of speech recognition in fighter aircraft. Of particular note are the U.S. program in speech recognition for the Advanced Fighter Technology Integration (AFTI)/F-16 aircraft (F-16 VISTA), the program in France on installing speech recognition systems on Mirage aircraft, and programs in the UK dealing with a variety of aircraft platforms. In these programs, speech recognizers have been operated successfully in fighter aircraft with applications including: setting radio frequencies, commanding an autopilot system, setting steer-point coordinates and weapons release parameters, and controlling flight displays.

Working with Swedish pilots flying in the JAS-39 Gripen cockpit, Englund (2004) found recognition deteriorated with increasing G-loads. It was also concluded that adaptation greatly improved the results in all cases and introducing models for breathing was shown to improve recognition scores significantly. Contrary to what might be expected, no effects of the broken English of the speakers were found. It was evident that spontaneous speech caused problems for the recognizer, as could be expected. A restricted vocabulary, and above all, a proper syntax, could thus be expected to improve recognition accuracy substantially.

The Eurofighter Typhoon currently in service with the UK RAF employs a speaker-dependent system, i.e. it requires each pilot to create a template. The system is not used for any safety critical or weapon critical tasks, such as weapon release or lowering of the undercarriage, but is used for a wide range of other cockpit functions. Voice commands are confirmed by visual and/or aural feedback. The system is seen as a major design feature in the reduction of pilot workload, and even allows the pilot to assign targets to himself with two simple voice commands or to any of his wingmen with only five commands.

Speaker independent systems are also being developed and are in testing for The F35 Lightning II (JSF) and the Aermacchi M346 lead in fighter trainer. These systems have produced word accuracies in excess of 98%.

Helicopters

The problems of achieving high recognition accuracy under stress and noise pertain strongly to the helicopter environment as well as to the fighter environment. The acoustic noise problem is actually more severe in the helicopter environment, not only because of the high noise levels but also because the helicopter pilot generally does not wear a facemask, which would reduce acoustic noise in the microphone. Substantial test and evaluation programs have been carried out in the past decade in speech recognition systems applications in helicopters, notably by the U.S. Army Avionics Research and Development Activity (AVRADA) and by the Royal Aerospace Establishment (RAE) in

the UK. Work in France has included speech recognition in the Puma helicopter. There has also been much useful work in Canada. Results have been encouraging, and voice applications have included: control of communication radios; setting of navigation systems; and control of an automated target handover system.

As in fighter applications, the overriding issue for voice in helicopters is the impact on pilot effectiveness. Encouraging results are reported for the AVRADA tests, although these represent only a feasibility demonstration in a test environment. Much remains to be done both in speech recognition and in overall speech recognition technology, in order to consistently achieve performance improvements in operational settings.

Battle management

Battle Management command centres generally require rapid access to and control of large, rapidly changing information databases. Commanders and system operators need to query these databases as conveniently as possible, in an eyes-busy environment where much of the information is presented in a display format. Human-machine interaction by voice has the potential to be very useful in these environments. A number of efforts have been undertaken to interface commercially available isolated-word recognizers into battle management environments. In one feasibility study speech recognition equipment was tested in conjunction with an integrated information display for naval battle management applications. Users were very optimistic about the potential of the system, although capabilities were limited.

Speech understanding programs sponsored by the Defense Advanced Research Projects Agency (DARPA) in the U.S. has focused on this problem of natural speech interface. Speech recognition efforts have focused on a database of continuous speech recognition (CSR), large-vocabulary speech which is designed to be representative of the naval resource management task. Significant advances in the state-of-the-art in CSR have been achieved, and current efforts are focused on integrating speech recognition and natural language processing to allow spoken language interaction with a naval resource management system.

Training air traffic controllers

Training for air traffic controllers (ATC) represents an excellent application for speech recognition systems. Many ATC training systems currently require a person to act as a "pseudo-pilot", engaging in a voice dialog with the trainee controller, which simulates the dialog which the controller would have to conduct with pilots in a real ATC situation. Speech recognition and synthesis techniques offer the potential to eliminate the need for a person to act as pseudo-pilot, thus reducing training and support personnel. In theory, Air controller tasks are also characterized by highly structured speech as the primary output of the controller, hence reducing the difficulty of the speech recognition task should be possible. In practice this is rarely the case. The FAA document 7110.65 details the phrases that should be used by air traffic controllers. While this document gives less than

150 examples of such phrases, the number of phrases supported by one of the simulation vendors speech recognition systems is in excess of 500,000.

The USAF, USMC, US Army,US Navy and FAA as well as a number of international ATC training organizations such as the Royal Australian Air Force and Civil Aviation Authorities in Italy, Brazil, Canada are currently using ATC simulators with speech recognition from a number of different vendors.

Telephony and other domains

ASR in the field of telephony is now commonplace and in the field of computer gaming and simulation is becoming more widespread. Despite the high level of integration with word processing in general personal computing, however, ASR in the field of document production has not seen the expected increases in use.

The improvement of mobile processor speeds made feasible the speech-enabled Symbian and Windows Mobile Smartphones. Speech is used mostly as a part of User Interface, for creating pre-defined or custom speech commands. Leading software vendors in this field are: Microsoft Corporation (Microsoft Voice Command), Nuance Communications (Nuance Voice Control), Vito Technology (VITO Voice2Go), Speereo Software (Speereo Voice Translator), Digital Syphon (Sonic Messager appliance) and SVOX.

Further applications

- Automatic translation;
- Automotive speech recognition (e.g., Ford Sync);
- Telematics (e.g. vehicle Navigation Systems);
- Court reporting (Realtime Voice Writing);
- Hands-free computing: voice command recognition computer user interface;
- Home automation;
- Interactive voice response;
- Mobile telephony, including mobile email;
- Multimodal interaction;
- Pronunciation evaluation in computer-aided language learning applications;
- Robotics;
- Video games, with Tom Clancy's EndWar and Lifeline as working examples;
- Transcription (digital speech-to-text);
- Speech-to-text (transcription of speech into mobile text messages);
- Air Traffic Control Speech Recognition.

Performance

The performance of speech recognition systems is usually specified in terms of accuracy and speed. Accuracy is usually rated with word error rate (WER), whereas speed is measured with the real time factor. Other measures of accuracy include Single Word Error Rate (SWER) and Command Success Rate (CSR).

In 1982, Kurzweil Applied Intelligence and Dragon Systems released speech recognition products. By 1985, Kurzweil's software had a vocabulary of 1,000 words—if uttered one word at a time. Two years later, in 1987, its lexicon reached 20,000 words, entering the realm of human vocabularies, which range from 10,000 to 150,000 words. But recognition accuracy was only 10% in 1993. Two years later, the error rate crossed below 50%. Dragon Systems released "Naturally Speaking" in 1997 which recognized normal human speech. Progress mainly came from improved computer performance and larger source text databases. The Brown Corpus was the first major database available, containing several million words. In 2001, recognition accuracy reached its current plateau of 80%, no longer growing with data or computing power. In 2006, Google published a trillion-word corpus, while Carnegie Mellon University researchers found no significant increase in recognition accuracy.

Algorithms

Both acoustic modeling and language modeling are important parts of modern statistically-based speech recognition algorithms. Hidden Markov models (HMMs) are widely used in many systems. Language modeling has many other applications such as smart keyboard and document classification.

Hidden Markov models

Modern general-purpose speech recognition systems are based on Hidden Markov Models. These are statistical models which output a sequence of symbols or quantities. HMMs are used in speech recognition because a speech signal can be viewed as a piecewise stationary signal or a short-time stationary signal. In a short-time (e.g., 10 milliseconds)), speech can be approximated as a stationary process. Speech can be thought of as a Markov model for many stochastic purposes.

Another reason why HMMs are popular is because they can be trained automatically and are simple and computationally feasible to use. In speech recognition, the hidden Markov model would output a sequence of *n*-dimensional real-valued vectors (with *n* being a small integer, such as 10), outputting one of these every 10 milliseconds. The vectors would consist of cepstral coefficients, which are obtained by taking a Fourier transform of a short time window of speech and decorrelating the spectrum using a cosine transform, then taking the first (most significant) coefficients. The hidden Markov model will tend to have in each state a statistical distribution that is a mixture of diagonal covariance Gaussians which will give a likelihood for each observed vector. Each word, or (for more general speech recognition systems), each phoneme, will have a different output distribution; a hidden Markov model for a sequence of words or phonemes is made by concatenating the individual trained hidden Markov models for the separate words and phonemes.

Described above are the core elements of the most common, HMM-based approach to speech recognition. Modern speech recognition systems use various combinations of a number of standard techniques in order to improve results over the basic approach

described above. A typical large-vocabulary system would need context dependency for the phonemes (so phonemes with different left and right context have different realizations as HMM states); it would use cepstral normalization to normalize for different speaker and recording conditions; for further speaker normalization it might use vocal tract length normalization (VTLN) for male-female normalization and maximum likelihood linear regression (MLLR) for more general speaker adaptation. The features would have so-called delta and delta-delta coefficients to capture speech dynamics and in addition might use heteroscedastic linear discriminant analysis (HLDA); or might skip the delta and delta-delta coefficients and use splicing and an LDA-based projection followed perhaps by heteroscedastic linear discriminant analysis or a global semitied covariance transform (also known as maximum likelihood linear transform, or MLLT). Many systems use so-called discriminative training techniques which dispense with a purely statistical approach to HMM parameter estimation and instead optimize some classification-related measure of the training data. Examples are maximum mutual information (MMI), minimum classification error (MCE) and minimum phone error (MPE).

Decoding of the speech (the term for what happens when the system is presented with a new utterance and must compute the most likely source sentence) would probably use the Viterbi algorithm to find the best path, and here there is a choice between dynamically creating a combination hidden Markov model which includes both the acoustic and language model information, or combining it statically beforehand (the finite state transducer, or FST, approach).

Dynamic time warping (DTW)-based speech recognition

Dynamic time warping is an approach that was historically used for speech recognition but has now largely been displaced by the more successful HMM-based approach. Dynamic time warping is an algorithm for measuring similarity between two sequences which may vary in time or speed. For instance, similarities in walking patterns would be detected, even if in one video the person was walking slowly and if in another they were walking more quickly, or even if there were accelerations and decelerations during the course of one observation. DTW has been applied to video, audio, and graphics – indeed, any data which can be turned into a linear representation can be analyzed with DTW.

A well known application has been automatic speech recognition, to cope with different speaking speeds. In general, it is a method that allows a computer to find an optimal match between two given sequences (e.g. time series) with certain restrictions, i.e. the sequences are "warped" non-linearly to match each other. This sequence alignment method is often used in the context of hidden Markov models.

Further information

Popular speech recognition conferences held each year or two include SpeechTEK and SpeechTEK Europe, ICASSP, Eurospeech/ICSLP (now named Interspeech) and the IEEE ASRU. Conferences in the field of Natural language processing, such as ACL,

NAACL, EMNLP, and HLT, are beginning to include papers on speech processing. Important journals include the IEEE Transactions on Speech and Audio Processing (now named IEEE Transactions on Audio, Speech and Language Processing), Computer Speech and Language, and Speech Communication. Books like "Fundamentals of Speech Recognition" by Lawrence Rabiner can be useful to acquire basic knowledge but may not be fully up to date (1993). Another good source can be "Statistical Methods for Speech Recognition" by Frederick Jelinek and "Spoken Language Processing (2001)" by Xuedong Huang etc. More up to date is "Computer Speech", by Manfred R. Schroeder, second edition published in 2004. The recently updated textbook of "Speech and Language Processing (2008)" by Jurafsky and Martin presents the basics and the state of the art for ASR. A good insight into the techniques used in the best modern systems can be gained by paying attention to government sponsored evaluations such as those organised by DARPA (the largest speech recognition-related project ongoing as of 2007 is the GALE project, which involves both speech recognition and translation components).

In terms of freely available resources, Carnegie Mellon University's SPHINX toolkit is one place to start to both learn about speech recognition and to start experimenting. Another resource (free as in free beer, not free software) is the HTK book (and the accompanying HTK toolkit). The AT&T libraries GRM library, and DCD library are also general software libraries for large-vocabulary speech recognition.

A useful review of the area of robustness in ASR is provided by Junqua and Haton (1995).

People with disabilities

People with disabilities can benefit from speech recognition programs. Speech recognition is especially useful for people who have difficulty using their hands, ranging from mild repetitive stress injuries to involved disabilities that preclude using conventional computer input devices. In fact, people who used the keyboard a lot and developed RSI became an urgent early market for speech recognition. Speech recognition is used in deaf telephony, such as voicemail to text, relay services, and captioned telephone. Individuals with learning disabilities who have problems with thought-to-paper communication (essentially they think of an idea but it is processed incorrectly causing it to end up differently on paper) can benefit from the software.

Current research and funding

Measuring progress in speech recognition performance is difficult and controversial. Some speech recognition tasks are much more difficult than others. Word error rates on some tasks are less than 1%. On others they can be as high as 50%. Sometimes it even appears that performance is going backwards as researchers undertake harder tasks that have higher error rates.

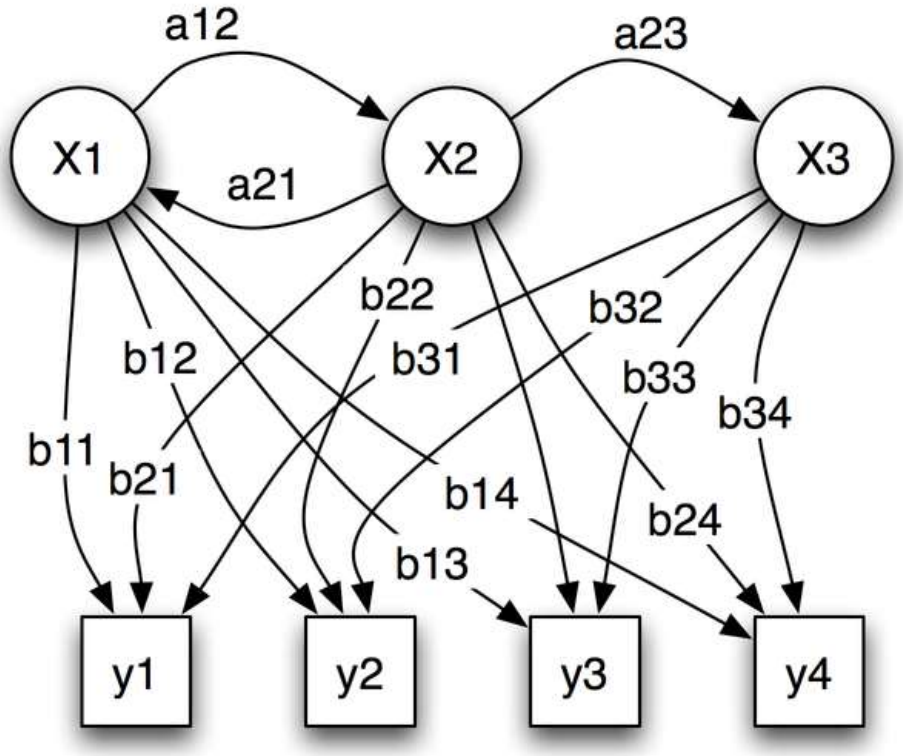
Because progress is slow and is difficult to measure, there is some perception that performance has plateaued and that funding has dried up or shifted priorities. Such perceptions are not new. In 1969, John Pierce wrote an open letter that did cause much funding to dry up for several years. In 1993 there was a strong feeling that performance had plateaued and there were workshops dedicated to the issue. However, in the 1990s funding continued more or less uninterrupted and performance continued slowly but steadily to improve.

For the past thirty years, speech recognition research has been characterized by the steady accumulation of small incremental improvements. There has also been a trend continually to change focus to more difficult tasks due both to progress in speech recognition performance and to the availability of faster computers. In particular, this shifting to more difficult tasks has characterized DARPA funding of speech recognition since the 1980s. In the last decade it has continued with the EARS project, which undertook recognition of Mandarin and Arabic in addition to English, and the GALE project, which focused solely on Mandarin and Arabic and required translation simultaneously with speech recognition.

Commercial research and other academic research also continue to focus on increasingly difficult problems. One key area is to improve robustness of speech recognition performance, not just robustness against noise but robustness against any condition that causes a major degradation in performance. Another key area of research is focused on an opportunity rather than a problem. This research attempts to take advantage of the fact that in many applications there is a large quantity of speech data available, up to millions of hours. It is too expensive to have humans transcribe such large quantities of speech, so the research focus is on developing new methods of machine learning that can effectively utilize large quantities of unlabeled data. Another area of research is better understanding of human capabilities and to use this understanding to improve machine recognition performance.

Chapter- 2

Hidden Markov Model



Probabilistic parameters of a hidden Markov model (example)
x — states
y — possible observations
a — state transition probabilities
b — output probabilities

A **hidden Markov model** (**HMM**) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (*hidden*) states. An HMM can be considered as the simplest dynamic Bayesian network.

In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a *hidden* Markov model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. Note that the adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model; even if the model parameters are known exactly, the model is still 'hidden'.

Hidden Markov models are especially known for their application in temporal pattern recognition such as speech, handwriting, gesture recognition, part-of-speech tagging, musical score following, partial discharges and bioinformatics.

A hidden Markov model can be considered a generalization of a mixture model where the hidden variables (or latent variables), which control the mixture component to be selected for each observation, are related through a Markov process rather than independent of each other.

Description in terms of urns

In its discrete form, a hidden Markov process can be visualized as a generalization of the familiar Urn problem. For instance, from Rabiner 1989: A genie is in a room that is not visible to the researcher. It is drawing balls labeled y1, y2, y3, ... from the urns X1, X2, X3, ... in that room and putting the balls on a conveyor belt, where the researcher can observe the sequence of the balls but not the sequence of urns from which they were chosen. The genie has some procedure to choose urns; the choice of the urn for the n-th ball depends upon only a random number and the choice of the urn for the $(n-1)$-th ball. Because the choice of urn does not directly depend on the urns further previous, this is called a Markov process.

Because the Markov process itself cannot be observed, and only the sequence of labeled balls can be observed, this arrangement is called a "hidden Markov process". This is illustrated by the lower part of the diagram above, where one can see that balls y1, y2, y3, y4 can be drawn at each state. Even if the researcher knows the composition of the urns and has just observed a sequence of three balls, *e.g.* y1, y1 and y1 on the conveyor belt, the researcher still cannot be sure from which urn (*i.e.*, at which state) the genie has drawn the third ball. However, the researcher can work out other details, such as the identity of the urn the genie is most likely to have drawn the third ball from.

Architecture of a hidden Markov model

The diagram below shows the general architecture of an instantiated HMM. Each oval shape represents a random variable that can adopt any of a number of values. The random variable $x(t)$ is the hidden state at time t (with the model from the above diagram, $x(t) \in \{ x_1, x_2, x_3 \}$). The random variable $y(t)$ is the observation at time t

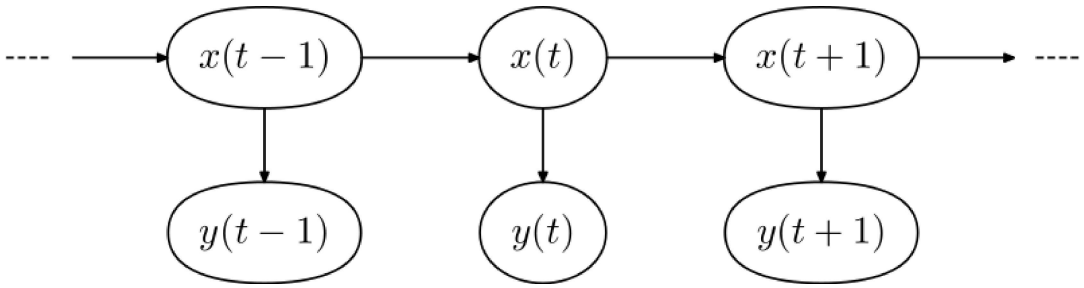
($y(t) \in \{ y_1, y_2, y_3, y_4 \}$). The arrows in the diagram (often called a trellis diagram) denote conditional dependencies.

From the diagram, it is clear that the conditional probability distribution of the hidden variable $x(t)$ at time t, given the values of the hidden variable x at all times, depends *only* on the value of the hidden variable $x(t - 1)$: the values at time $t - 2$ and before have no influence. This is called the Markov property. Similarly, the value of the observed variable $y(t)$ only depends on the value of the hidden variable $x(t)$ (both at time t).

In the standard type of hidden Markov model considered here, the state space of the hidden variables is discrete, while the observations themselves can either be discrete (typically generated from a categorical distribution) or continuous (typically from a Gaussian distribution). The parameters of a hidden Markov model are of two types, *transition probabilities* and *emission probabilities* (also known as *output probabilities*). The transition probabilities control the way the hidden state at time t is chosen given the hidden state at time $t - 1$.

The hidden state space is assumed to consist of one of N possible values, modeled as a categorical distribution. This means that for each of the N possible states that a hidden variable at time t can be in, there is a transition probability from this state to each of the N possible states of the hidden variable at time $t + 1$, for a total of N^2 transition probabilities. (Note, however, that the set of transition probabilities for transitions from any given state must sum to 1, meaning that any one transition probability can be determined once the others are known, leaving a total of $N(N - 1)$ transition parameters.)

In addition, for each of the N possible states, there is a set of emission probabilities governing the distribution of the observed variable at a particular time given the state of the hidden variable at that time. The size of this set depends on the nature of the observed variable. For example, if the observed variable is discrete with M possible values, governed by a categorical distribution, there will be $M - 1$ separate parameters, for a total of $N(M - 1)$ emission parameters over all hidden states. On the other hand, if the observed variable is an M-dimensional vector distributed according to an arbitrary multivariate Gaussian distribution, there will be M parameters controlling the means and $M(M + 1) / 2$ parameters controlling the covariance matrix, for a total of $N(M + \frac{M(M+1)}{2}) = NM(M+3)/2 = O(NM^2)$ emission parameters. (In such a case, unless the value of M is small, it may be more practical to restrict the nature of the covariances between individual elements of the observation vector, e.g. by assuming that the elements are independent of each other, or less restrictively, are independent of all but a fixed number of adjacent elements.)



Mathematical description of a hidden Markov model

General description

A basic, non-Bayesian hidden Markov model can be described as follows:

$$
\begin{array}{lcl}
N & = & \text{number of states} \\
T & = & \text{number of observations} \\
\theta_{i=1\ldots N} & = & \text{emission parameter for an observation associated with state } i \\
\phi_{i=1\ldots N, j=1\ldots N} & = & \text{probability of transition from state } i \text{ to state } j \\
\boldsymbol{\phi}_{i=1\ldots N} & = & N\text{-dimensional vector, composed of } \phi_{i,1\ldots N}\text{; must sum to 1} \\
x_{t=1\ldots T} & = & \text{state of observation at time } t \\
y_{t=1\ldots T} & = & \text{observation at time } t \\
F(y|\theta) & = & \text{probability distribution of an observation, parametrized on } \theta \\
x_{t=2\ldots T} & \sim & \text{Categorical}(\boldsymbol{\phi}_{x_{t-1}}) \\
y_{t=1\ldots T} & \sim & F(\theta_{x_t})
\end{array}
$$

Note that, in the above model (and also the one below), the prior distribution of the initial state x_1 is not specified. Typical learning models correspond to assuming a discrete uniform distribution over possible states (i.e. no particular prior distribution is assumed).

In a Bayesian setting, all parameters are associated with random variables, as follows:

$$
\begin{array}{lcl}
N, T & = & \text{as above} \\
\theta_{i=1\ldots N}, \phi_{i=1\ldots N, j=1\ldots N}, \boldsymbol{\phi}_{i=1\ldots N} & = & \text{as above} \\
x_{t=1\ldots T}, y_{t=1\ldots T}, F(y|\theta) & = & \text{as above} \\
\alpha & = & \text{shared hyperparameter for emission parameters} \\
\beta & = & \text{shared hyperparameter for transition parameters} \\
H(\theta|\alpha) & = & \text{prior probability distribution of emission parameters, parametrized on } \alpha \\
\theta_{i=1\ldots N} & \sim & H(\alpha) \\
\boldsymbol{\phi}_{i=1\ldots N} & \sim & \text{Symmetric-Dirichlet}_N(\beta) \\
x_{t=2\ldots T} & \sim & \text{Categorical}(\boldsymbol{\phi}_{x_{i-t}}) \\
y_{t=1\ldots T} & \sim & F(\theta_{x_t})
\end{array}
$$

These characterizations use *F* and *H* to describe arbitrary distributions over observations and parameters, respectively. Typically *H* will be the conjugate prior of *F*. The two most common choices of *F* are Gaussian and categorical; see below.

Compared with a simple mixture model

As mentioned above, the distribution of each observation in a hidden Markov model is a mixture density, with the states of the HMM corresponding to mixture components. It is useful to compare the above characterizations for an HMM with the corresponding characterizations, of a mixture model, using the same notation.

A non-Bayesian mixture model:

$$
\begin{array}{lcl}
N & = & \text{number of mixture components} \\
T & = & \text{number of observations} \\
\theta_{i=1\ldots N} & = & \text{parameter of distribution of observation associated with component } i \\
\phi_{i=1\ldots N} & = & \text{mixture weight, i.e. prior probability of a particular component } i \\
\boldsymbol{\phi} & = & N\text{-dimensional vector composed of all the individual } \phi_{1\ldots N}\text{; must sum to 1} \\
x_{i=1\ldots T} & = & \text{component of observation } i \\
y_{i=1\ldots T} & = & \text{observation } i \\
F(y|\theta) & = & \text{probability distribution of an observation, parametrized on } \theta \\
x_{i=1\ldots T} & \sim & \text{Categorical}(\boldsymbol{\phi}) \\
y_{i=1\ldots T} & \sim & F(\theta_{x_i})
\end{array}
$$

A Bayesian mixture model:

$$
\begin{array}{lcl}
N, T & = & \text{as above} \\
\theta_{i=1\ldots N}, \phi_{i=1\ldots N}, \boldsymbol{\phi} & = & \text{as above} \\
x_{i=1\ldots T}, y_{i=1\ldots T}, F(y|\theta) & = & \text{as above} \\
\alpha & = & \text{shared hyperparameter for component parameters} \\
\beta & = & \text{shared hyperparameter for mixture weights} \\
H(\theta|\alpha) & = & \text{prior probability distribution of component parameters, parametrized on } \alpha \\
\theta_{i=1\ldots N} & \sim & H(\alpha) \\
\boldsymbol{\phi} & \sim & \text{Symmetric-Dirichlet}_N(\beta) \\
x_{i=1\ldots T} & \sim & \text{Categorical}(\boldsymbol{\phi}) \\
y_{i=1\ldots T} & \sim & F(\theta_{x_i})
\end{array}
$$

Examples of HMMs

The following mathematical descriptions are fully written out and explained, for ease of implementation.

A typical non-Bayesian HMM with Gaussian observations looks like this:

$$
\begin{array}{lcl}
N & = & \text{number of states} \\
T & = & \text{number of observations} \\
\phi_{i=1\ldots N, j=1\ldots N} & = & \text{probability of transition from state } i \text{ to state } j \\
\boldsymbol{\phi}_{i=1\ldots N} & = & N\text{-dimensional vector, composed of } \phi_{i,1\ldots N}\text{; must sum to 1} \\
\mu_{i=1\ldots N} & = & \text{mean of observations associated with state } i \\
\sigma^2_{i=1\ldots N} & = & \text{variance of observations associated with state } i \\
x_{t=1\ldots T} & = & \text{state of observation at time } t \\
y_{t=1\ldots T} & = & \text{observation at time } t \\
x_{t=2\ldots T} & \sim & \text{Categorical}(\boldsymbol{\phi}_{x_{t-1}}) \\
y_{t=1\ldots T} & \sim & \mathcal{N}(\mu_{x_t}, \sigma^2_{x_t})
\end{array}
$$

A typical Bayesian HMM with Gaussian observations looks like this:

$$
\begin{array}{lcl}
N & = & \text{number of states} \\
T & = & \text{number of observations} \\
\phi_{i=1\ldots N, j=1\ldots N} & = & \text{probability of transition from state } i \text{ to state } j \\
\boldsymbol{\phi}_{i=1\ldots N} & = & N\text{-dimensional vector, composed of } \phi_{i,1\ldots N}\text{; must sum to 1} \\
\mu_{i=1\ldots N} & = & \text{mean of observations associated with state } i \\
\sigma^2_{i=1\ldots N} & = & \text{variance of observations associated with state } i \\
x_{t=1\ldots T} & = & \text{state of observation at time } t \\
y_{t=1\ldots T} & = & \text{observation at time } t \\
\beta & = & \text{concentration hyperparameter controlling the density of the transition matrix} \\
\mu_0, \lambda & = & \text{shared hyperparameters of the means for each state} \\
\nu, \sigma_0^2 & = & \text{shared hyperparameters of the variances for each state} \\
\boldsymbol{\phi}_{i=1\ldots N} & \sim & \text{Symmetric-Dirichlet}_N(\beta) \\
x_{t=2\ldots T} & \sim & \text{Categorical}(\boldsymbol{\phi}_{x_{t-1}}) \\
\mu_{i=1\ldots N} & \sim & \mathcal{N}(\mu_0, \lambda\sigma_i^2) \\
\sigma^2_{i=1\ldots N} & \sim & \text{Inverse-Gamma}(\nu, \sigma_0^2) \\
y_{t=1\ldots T} & \sim & \mathcal{N}(\mu_{x_t}, \sigma^2_{x_t})
\end{array}
$$

A typical non-Bayesian HMM with categorical observations looks like this:

$$
\begin{array}{lcl}
N & = & \text{number of states} \\
T & = & \text{number of observations} \\
\phi_{i=1\ldots N, j=1\ldots N} & = & \text{probability of transition from state } i \text{ to state } j \\
\boldsymbol{\phi}_{i=1\ldots N} & = & N\text{-dimensional vector, composed of } \phi_{i,1\ldots N}\text{; must sum to 1} \\
V & = & \text{dimension of categorical observations, e.g. size of word vocabulary} \\
\theta_{i=1\ldots N, j=1\ldots V} & = & \text{probability for state } i \text{ of observing the } j\text{th item} \\
\boldsymbol{\theta}_{i=1\ldots N} & = & V\text{-dimensional vector, composed of } \theta_{i,1\ldots V}\text{; must sum to 1} \\
x_{t=1\ldots T} & = & \text{state of observation at time } t \\
y_{t=1\ldots T} & = & \text{observation at time } t \\
x_{t=2\ldots T} & \sim & \text{Categorical}(\boldsymbol{\phi}_{x_{t-1}}) \\
y_{t=1\ldots T} & \sim & \text{Categorical}(\boldsymbol{\theta}_{x_t})
\end{array}
$$

A typical Bayesian HMM with categorical observations looks like this:

$$
\begin{array}{lcl}
N & = & \text{number of states} \\
T & = & \text{number of observations} \\
\phi_{i=1\ldots N, j=1\ldots N} & = & \text{probability of transition from state } i \text{ to state } j \\
\boldsymbol{\phi}_{i=1\ldots N} & = & N\text{-dimensional vector, composed of } \phi_{i,1\ldots N}\text{; must sum to 1} \\
V & = & \text{dimension of categorical observations, e.g. size of word vocabulary} \\
\theta_{i=1\ldots N, j=1\ldots V} & = & \text{probability for state } i \text{ of observing the } j\text{th item} \\
\boldsymbol{\theta}_{i=1\ldots N} & = & V\text{-dimensional vector, composed of } \theta_{i,1\ldots V}\text{; must sum to 1} \\
x_{t=1\ldots T} & = & \text{state of observation at time } t \\
y_{t=1\ldots T} & = & \text{observation at time } t \\
\alpha & = & \text{shared concentration hyperparameter of } \boldsymbol{\theta} \text{ for each state} \\
\beta & = & \text{concentration hyperparameter controlling the density of the transition matrix} \\
\boldsymbol{\phi}_{i=1\ldots N} & \sim & \text{Symmetric-Dirichlet}_N(\beta) \\
\boldsymbol{\theta}_{1\ldots V} & \sim & \text{Symmetric-Dirichlet}_V(\alpha) \\
x_{t=2\ldots T} & \sim & \text{Categorical}(\boldsymbol{\phi}_{x_{t-1}}) \\
y_{t=1\ldots T} & \sim & \text{Categorical}(\boldsymbol{\theta}_{x_t})
\end{array}
$$

Note that in the above Bayesian characterizations, β (a concentration parameter) controls the density of the transition matrix. That is, with a high value of β (significantly above 1),

the probabilities controlling the transition out of a particular state will all be similar, meaning there will be a significantly probability of transitioning to any of the other states. In other words, the path followed by the Markov chain of hidden states will be highly random. With a low value of β (significantly below 1), only a small number of the possible transitions out of a given state will have significant probability, meaning that the path followed by the hidden states will be somewhat predictable.

A two-level Bayesian HMM

An alternative for the above two Bayesian examples would be to add another level of prior parameters for the transition matrix. That is, replace the lines

$$\begin{array}{lcl} \beta & = & \text{concentration hyperparameter controlling the density of the transition matrix} \\ \boldsymbol{\phi}_{i=1\ldots N} & \sim & \text{Symmetric-Dirichlet}_N(\beta) \end{array}$$

with the following:

$$\begin{array}{lcl} \gamma & = & \text{concentration hyperparameter controlling how many states are intrinsically likely} \\ \beta & = & \text{concentration hyperparameter controlling the density of the transition matrix} \\ \boldsymbol{\eta} & = & N\text{-dimensional vector of probabilities, specifying the intrinsic probability of a given state} \\ \boldsymbol{\eta} & \sim & \text{Symmetric-Dirichlet}_N(\gamma) \\ \boldsymbol{\phi}_{i=1\ldots N} & \sim & \text{Dirichlet}_N(\beta N \boldsymbol{\eta}) \end{array}$$

What this means is the following:

1. $\boldsymbol{\eta}$ is a probability distribution over states, specifying which states are inherently likely. The greater the probability of a given state in this vector, the more likely is a transition to that state (regardless of the starting state).
2. γ controls the density of $\boldsymbol{\eta}$. Values significantly above 1 cause a dense vector where all states will have similar prior probabilities. Values significantly below 1 cause a sparse vector where only a few states are inherently likely (have prior probabilities significantly above 0).
3. β controls the density of the transition matrix, or more specifically, the density of the N different probability vectors $\boldsymbol{\phi}_{i=1\ldots N}$ specifying the probability of transitions out of state i to any other state.

Imagine that the value of β is significantly above 1. Then the different $\boldsymbol{\phi}$ vectors will be dense, i.e. the probability mass will be spread out fairly evenly over all states. However, to the extent that this mass is unevenly spread, $\boldsymbol{\eta}$ controls which states are likely to get more mass than others.

Now, imagine instead that β is significantly below 1. This willl make the $\boldsymbol{\phi}$ vectors sparse, i.e. almost all the probability mass is distributed over a small number of states, and for the rest, a transition to that state will be very unlikely. Notice that there are different $\boldsymbol{\phi}$ vectors for each starting state, and so even if all the vectors are sparse, different vectors may distribute the mass to different ending states. However, for all of the vectors, $\boldsymbol{\eta}$ controls which ending states are likely to get mass assigned to them. For

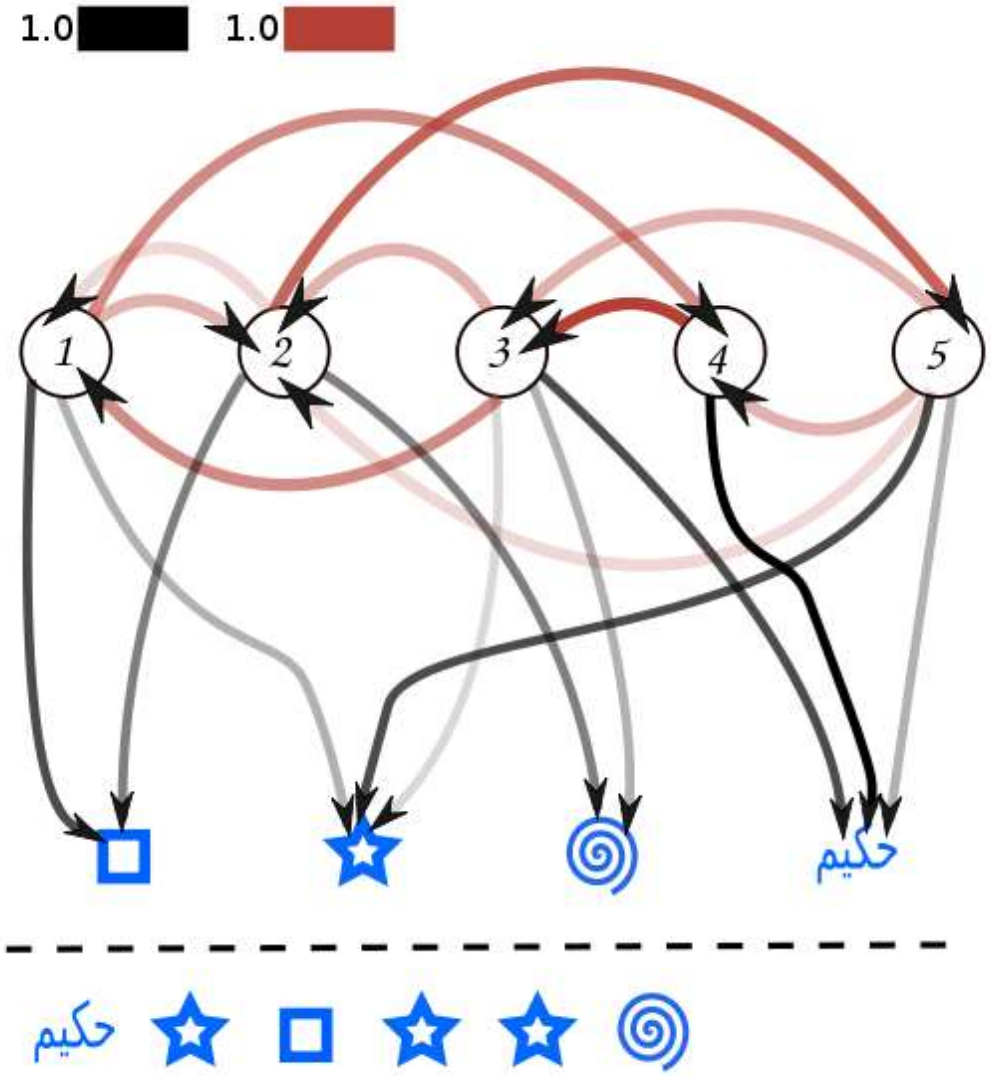
example, if β is 0.1, then each ϕ will be sparse and, for any given starting state *i*, the set of states $\mathbf{J}_i$ to which transitions are likely to occur will be very small, typically having only one or two members. Now, if the probabilities in η are all the same (or equivalently, one of the above models without η is used), then for different *i*, there will be different states in the corresponding $\mathbf{J}_i$, so that all states are equally likely to occur in any given $\mathbf{J}_i$. On the other hand, if the values in η are unbalanced, so that one state has a much higher probability than others, almost all $\mathbf{J}_i$ will contain this state; hence, regardless of the starting state, transitions will nearly always occur to this given state.

Hence, a two-level model such as just described allows independent control over (1) the overall density of the transition matrix, and (2) the density of states to which transitions are likely (i.e. the density of the prior distribution of states in any particular hidden variable x_i). In both cases this is done while still assuming ignorance over which particular states are more likely than others. If it is desired to inject this information into the model, the probability vector η can be directly specified; or, if there is less certainty about these relative probabilities, a non-symmetric Dirichlet distribution can be used as the prior distribution over η. That is, instead of using a symmetric Dirichlet distribution with a single parameter γ (or equivalently, a general Dirichlet with a vector all of whose values are equal to γ), use a general Dirichlet with values that are variously greater or less than γ, according to which state is more or less preferred.

Learning

The parameter learning task in HMMs is to find, given an output sequence or a set of such sequences, the best set of state transition and output probabilities. The task is usually to derive the maximum likelihood estimate of the parameters of the HMM given the set of output sequences. No tractable algorithm is known for solving this problem exactly, but a local maximum likelihood can be derived efficiently using the Baum-Welch algorithm or the Baldi-Chauvin algorithm. The Baum-Welch algorithm is an example of a forward-backward algorithm, and is a special case of the Expectation-maximization algorithm.

Inference



The state transition and output probabilities of an HMM are indicated by the line opacity in the upper part of the diagram. Given that we have observed the output sequence in the lower part of the diagram, we may be interested in the most likely sequence of states that could have produced it. Based on the arrows that are present in the diagram, the following state sequences are candidates:

5 3 2 5 3 2
4 3 2 5 3 2
3 1 2 5 3 2

We can find the most likely sequence by evaluating the joint probability of both the state sequence and the observations for each case (simply by multiplying the probability values, which here correspond to the opacities of the arrows involved). In general, this type of problem (i.e. finding the most likely explanation for an observation sequence) can be solved efficiently.

Several inference problems are associated with hidden Markov models, as outlined below.

Filtering

The task is to compute, given the model's parameters and a sequence of observations, the distribution over hidden states at the end of the sequence, i.e. to compute $P(x(t) \mid y(1), \ldots, y(t))$. This problem can be handled efficiently using the forward algorithm.

Probability of an observed sequence

The task is to compute, given the parameters of the model, the probability of a particular output sequence. This requires summation over all possible state sequences:

The probability of observing a sequence

$$Y = y(0), y(1), \ldots, y(L-1)$$

of length *L* is given by

$$P(Y) = \sum_X P(Y \mid X) P(X),$$

where the sum runs over all possible hidden-node sequences

$$X = x(0), x(1), \ldots, x(L-1).$$

Applying the principle of dynamic programming, this problem, too, can be handled efficiently using the forward algorithm.

Most likely explanation

The task is to compute, given the parameters of the model and a particular output sequence, the state sequence that is most likely to have generated that output sequence. This requires finding a maximum over all possible state sequences, but can similarly be solved efficiently by the Viterbi algorithm.

Smoothing

The task is to compute, given the parameters of the model and a particular output sequence up to time *t*, the probability distribution over hidden states for a point in time in the past, i.e. to compute $P(x(k) \mid y(1), \ldots, y(t))$ for some $k < t$. The forward-backward algorithm is an efficient method for computing the smoothed values for all hidden state variables.

Statistical Significance

For some of the above problems, it may also be interesting to ask about statistical significance. What is the probability that a sequence drawn from some null distribution will have an HMM probability (in the case of the forward algorithm) or a maximum state sequence probability (in the case of the Viterbi algorithm) at least as large as that of a particular output sequence? When an HMM is used to evaluate the relevance of a hypothesis for a particular output sequence, the statistical significance indicates the false positive rate associated with accepting the hypothesis for the output sequence.

A concrete example

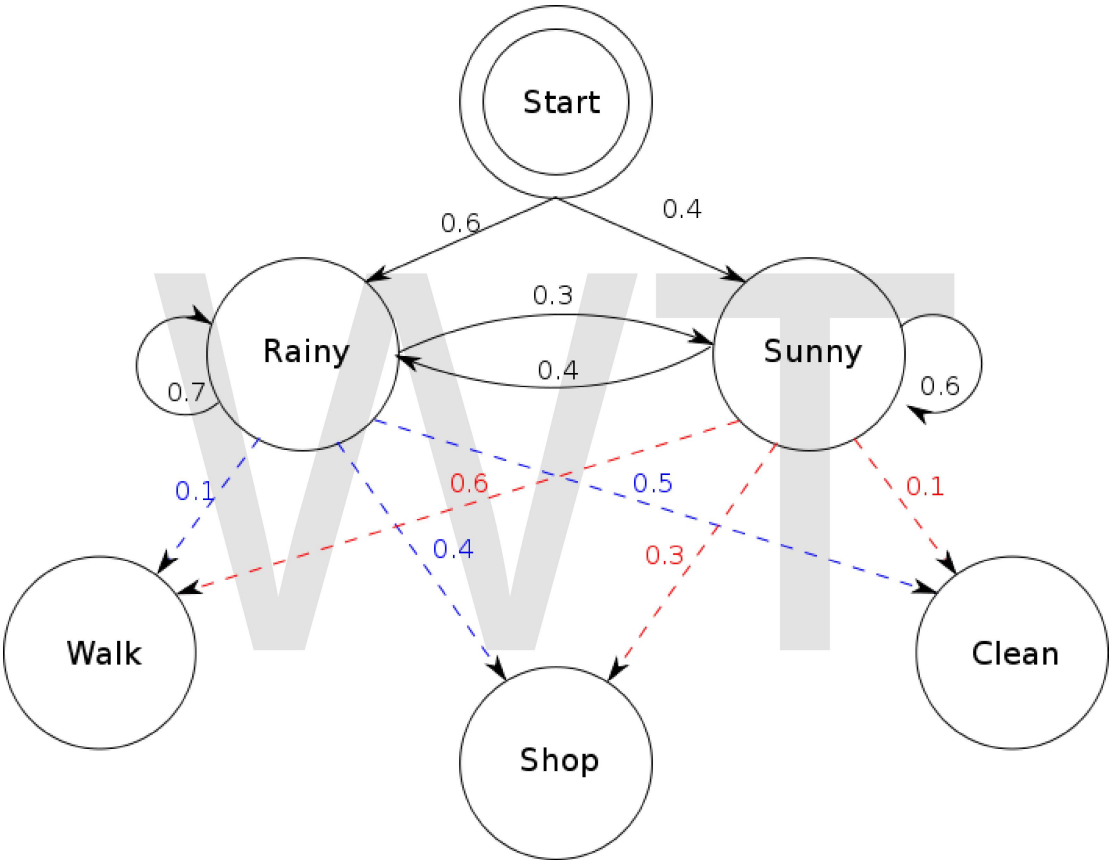
Consider two friends, Alice and Bob, who live far apart from each other and who talk together daily over the telephone about what they did that day. Bob is only interested in three activities: walking in the park, shopping, and cleaning his apartment. The choice of what to do is determined exclusively by the weather on a given day. Alice has no definite information about the weather where Bob lives, but she knows general trends. Based on what Bob tells her he did each day, Alice tries to guess what the weather must have been like.

Alice believes that the weather operates as a discrete Markov chain. There are two states, "Rainy" and "Sunny", but she cannot observe them directly, that is, they are *hidden* from her. On each day, there is a certain chance that Bob will perform one of the following activities, depending on the weather: "walk", "shop", or "clean". Since Bob tells Alice about his activities, those are the *observations*. The entire system is that of a hidden Markov model (HMM).

Alice knows the general weather trends in the area, and what Bob likes to do on average. In other words, the parameters of the HMM are known. They can be written down in the Python programming language:

```
states = ('Rainy', 'Sunny')

observations = ('walk', 'shop', 'clean')

start_probability = {'Rainy': 0.6, 'Sunny': 0.4}

transition_probability = {
   'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
   'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},
   }

emission_probability = {
   'Rainy' : {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
   'Sunny' : {'walk': 0.6, 'shop': 0.3, 'clean': 0.1},
   }
```

In this piece of code, `start_probability` represents Alice's belief about which state the HMM is in when Bob first calls her (all she knows is that it tends to be rainy on average). The particular probability distribution used here is not the equilibrium one, which is (given the transition probabilities) approximately `{'Rainy': 0.57, 'Sunny': 0.43}`. The `transition_probability` represents the change of the weather in the underlying Markov chain. In this example, there is only a 30% chance that tomorrow will be sunny if today is rainy. The `emission_probability` represents how likely Bob is to perform a certain activity on each day. If it is rainy, there is a 50% chance that he is cleaning his apartment; if it is sunny, there is a 60% chance that he is outside for a walk.



This example is further elaborated in the Viterbi algorithm page.

Applications of hidden Markov models

HMMs can be applied in many fields where the goal is to recover a data sequence that is not immediately observable (but other data that depends on the sequence is). Common applications include:

- Cryptanalysis
- Speech recognition
- Part-of-speech tagging
- Machine translation

- Partial discharge
- Gene prediction
- Alignment of bio-sequences
- Activity recognition

History

Hidden Markov Models were first described in a series of statistical papers by Leonard E. Baum and other authors in the second half of the 1960s. One of the first applications of HMMs was speech recognition, starting in the mid-1970s.

In the second half of the 1980s, HMMs began to be applied to the analysis of biological sequences , in particular DNA. Since then, they have become ubiquitous in the field of bioinformatics.

Types of hidden Markov models

Hidden Markov models can model complex Markov processes where the states emit the observations according to some probability distribution. One such example of distribution is Gaussian distribution, in such a Hidden Markov Model the states output is represented by a Gaussian distribution.

Moreover it could represent even more complex behavior when the output of the states is represented as mixture of two or more Gaussians, in which case the probability of generating an observation is the product of the probability of first selecting one of the Gaussians and the probability of generating that observation from that Gaussian.

Extensions

In the hidden Markov models considered above, the state space of the hidden variables is discrete, while the observations themselves can either be discrete (typically generated from a categorical distribution) or continuous (typically from a Gaussian distribution). Hidden Markov models can also be generalized to allow continuous state spaces. Examples of such models are those where the Markov process over hidden variables is a linear dynamical system, with a linear relationship among related variables and where all hidden and observed variables follow a Gaussian distribution. In simple cases, such as the linear dynamical system just , exact inference is tractable (in this case, using the Kalman filter); however, in general, exact inference in HMMs with continuous latent variables in infeasible, and approximate methods must be used, such as the extended Kalman filter or the particle filter.

Hidden Markov models are generative models, in which the joint distribution of observations and hidden states, or equivalently both the prior distribution of hidden states (the *transition probabilities*) and conditional distribution of observations given states (the *emission probabilities*), is modeled. The above algorithms implicitly assume a uniform

prior distribution over the transition probabilities. However, it is also possible to create hidden Markov models with other types of prior distributions. An obvious candidate, given the categorical distribution of the transition probabilities, is the Dirichlet distribution, which is the conjugate prior distribution of the categorical distribution. Typically, a symmetric Dirichlet distribution is chosen, reflecting ignorance about which states are inherently more likely than others. The single parameter of this distribution (termed the *concentration parameter*) controls the relative density or sparseness of the resulting transition matrix. A choice of 1 yields a uniform distribution. Values greater than 1 produce a dense matrix, in which the transition probabilities between pairs of states are likely to be nearly equal. Values less than 1 result in a sparse matrix in which, for each given source state, only a small number of destination states have non-negligible transition probabilities. It is also possible to use a two-level prior Dirichlet distribution, in which one Dirichlet distribution (the upper distribution) governs the parameters of another Dirichlet distribution (the lower distribution), which in turn governs the transition probabilities. The upper distribution governs the overall distribution of states, determining how likely each state is to occur; its concentration parameter determines the density or sparseness of states. Such a two-level prior distribution, where both concentration parameters are set to produce sparse distributions, might be useful for example in unsupervised part-of-speech tagging, where some parts of speech occur much more commonly than others; learning algorithms that assume a uniform prior distribution generally perform poorly on this task. The parameters of models of this sort, with non-uniform prior distributions, can be learned using Gibbs sampling or extended versions of the expectation-maximization algorithm.

An extension of the previously-described hidden Markov models with Dirichlet priors uses a Dirichlet process in place of a Dirichlet distribution. This type of model allows for an unknown and potentially infinite number of states. It is common to use a two-level Dirichlet process, similar to the previously-described model with two levels of Dirichlet distributions. Such a model is called a *hierarchical Dirichlet process hidden Markov model*, or *HDP-HMM* for short.

A different type of extension uses a discriminative model in place of the generative model of standard HMM's. This type of model directly models the conditional distribution of the hidden states given the observations, rather than modeling the joint distribution. An example of this model is the so-called *maximum entropy Markov model* (MEMM), which models the conditional distribution of the states using logistic regression (also known as a "maximum entropy model"). The advantage of this type of model is that arbitrary features (i.e. functions) of the observations can be modeled, allowing domain-specific knowledge of the problem at hand to be injected into the model. Models of this sort are not limited to modeling direct dependencies between a hidden state and its associated observation; rather, features of nearby observations, of combinations of the associated observation and nearby observations, or in fact of arbitrary observations at any distance from a given hidden state can be included in the process used to determine the value of a hidden state. Furthermore, there is no need for these features to be statistically independent of each other, as would be the case if such features were used in a generative model. Finally, arbitrary features over pairs of adjacent

hidden states can be used rather than simple transition probabilities. The disadvantages of such models are: (1) The types of prior distributions that can be placed on hidden states are severely limited; (2) It is not possible to predict the probability of seeing an arbitrary observation. This second limitation is often not an issue in practice, since many common usages of HMM's do not require such predictive probabilities.

A variant of the previously described discriminative model is the linear-chain conditional random field. This uses an undirected graphical model (aka Markov random field) rather than the directed graphical models of MEMM's and similar models. The advantage of this type of model is that it does not suffer from the so-called *label bias* problem of MEMM's, and thus may make more accurate predictions. The disadvantage is that training can be slower than for MEMM's.

Yet another variant is the *factorial hidden Markov model*, which allows for a single observation to be conditioned on the corresponding hidden variables of a set of *K* independent Markov chains, rather than a single Markov chain. Learning in such a model is difficult, as dynamic-programming techniques can no longer be used to find an exact solution; in practice, approximate techniques must be used.

All of the above models can be extended to allow for more distant dependencies among hidden states, e.g. allowing for a given state to be dependent on the previous two or three states rather than a single previous state; i.e. the transition probabilities are extended to encompass sets of three or four adjacent states (or in general *K* adjacent states). The disadvantage of such models is that dynamic-programming algorithms for training them have an $O(N^K T)$ running time, for *K* adjacent states and *T* total observations (i.e. a length-*T* Markov chain).

Chapter- 3

Machine Translation

Machine translation, sometimes referred to by the abbreviation **MT**, also called **computer-aided translation**, **machine-aided human translation MAHT** and **interactive translation**, is a sub-field of computational linguistics that investigates the use of computer software to translate text or speech from one natural language to another. At its basic level, MT performs simple substitution of words in one natural language for words in another, but that alone usually cannot produce a good translation of a text, because recognition of whole phrases and their closest counterparts in the target language is needed. Solving this problem with corpus and statistical techniques is a rapidly growing field that is leading to better translations, handling differences in linguistic typology, translation of idioms, and the isolation of anomalies.

Current machine translation software often allows for customisation by domain or profession (such as weather reports), improving output by limiting the scope of allowable substitutions. This technique is particularly effective in domains where formal or formulaic language is used. It follows that machine translation of government and legal documents more readily produces usable output than conversation or less standardised text.

Improved output quality can also be achieved by human intervention: for example, some systems are able to translate more accurately if the user has unambiguously identified which words in the text are names. With the assistance of these techniques, MT has proven useful as a tool to assist human translators and, in a very limited number of cases, can even produce output that can be used as is (e.g., weather reports).

History

The **history of machine translation** generally starts in the 1950s, although work can be found from earlier periods. The Georgetown experiment in 1954 involved fully automatic translation of more than sixty Russian sentences into English. The experiment was a great success and ushered in an era of significant funding for machine translation research. The

authors claimed that within three or five years, machine translation would be a solved problem.

However, the real progress was much slower, and after the ALPAC report in 1966, which found that the ten years of research had failed to fulfill the expectations, and funding was dramatically reduced. Starting in the late 1980s, as computational power increased and became less expensive, more interest began to be shown in statistical models for machine translation.

Today there is still no system that provides the holy-grail of "fully automatic high quality translation of unrestricted text" (FAHQUT). However, there are many programs now available that are capable of providing useful output within strict constraints; several of them are available online, such as Google Translate and the SYSTRAN system which powers AltaVista's (Yahoo's since May 9, 2008) BabelFish.

The beginning

The history of machine translation dates back to the seventeenth century, when philosophers such as Leibniz and Descartes put forward proposals for codes which would relate words between languages. All of these proposals remained theoretical, and none resulted in the development of an actual machine.

The first patents for "translating machines" were applied for in the mid 1930s. One proposal, by Georges Artsrouni was simply an automatic bilingual dictionary using paper tape. The other proposal, by Peter Troyanskii, a Russian, was more detailed. It included both the bilingual dictionary, and a method for dealing with grammatical roles between languages, based on Esperanto. The system was split up into three stages: the first was for a native-speaking editor in the sources language to organise the words into their logical forms and syntactic functions; the second was for the machine to "translate" these forms into the target language; and the third was for a native-speaking editor in the target language to normalise this output. His scheme remained unknown until the late 1950s, by which time computers were well-known.

The early years

The first proposals for machine translation using computers were put forward by Warren Weaver, a researcher at the Rockefeller Foundation, in his July, 1949 memorandum. These proposals were based on information theory, successes of code breaking during the second world war and speculation about universal underlying principles of natural language.

A few years after these proposals, research began in earnest at many universities in the United States. On 7 January 1954, the Georgetown-IBM experiment, the first public demonstration of an MT system, was held in New York at the head office of IBM. The demonstration was widely reported in the newspapers and received much public interest.

The system itself, however, was no more than what today would be called a "toy" system, having just 250 words and translating just 49 carefully selected Russian sentences into English — mainly in the field of chemistry. Nevertheless it encouraged the view that machine translation was imminent — and in particular stimulated the financing of the research, not just in the US but worldwide.

Early systems used large bilingual dictionaries and hand-coded rules for fixing the word order in the final output. This was eventually found to be too restrictive, and developments in linguistics at the time, for example generative linguistics and transformational grammar were proposed to improve the quality of translations.

During this time, operational systems were installed. The United States Air Force used a system produced by IBM and Washington University, while the Atomic Energy Commission in the United States and Euratom in Italy used a system developed at Georgetown University. While the quality of the output was poor, it nevertheless met many of the customers' needs, chiefly in terms of speed.

At the end of the 1950s, an argument was put forward by Yehoshua Bar-Hillel, a researcher asked by the US government to look into machine translation against the possibility of "Fully Automatic High Quality Translation" by machines. The argument is one of semantic ambiguity or double-meaning. Consider the following sentence:

> Little John was looking for his toy box. Finally he found it. The box was in the pen.

The word *pen* may have two meanings, the first meaning something you use to write with, the second meaning a container of some kind. To a human, the meaning is obvious, but he claimed that without a "universal encyclopedia" a machine would never be able to deal with this problem. Today, this type of semantic ambiguity can be solved by writing source texts for machine translation in a controlled language that uses a vocabulary in which each word has exactly one meaning.

The 1960s, the ALPAC report and the seventies

Research in the 1960s in both the Soviet Union and the United States concentrated mainly on the Russian-English language pair. Chiefly the objects of translation were scientific and technical documents, such as articles from scientific journals.

A great blow came to machine translation research in 1966 with the publication of the ALPAC report. The report was commissioned by the US government and performed by ALPAC, the Automatic Language Processing Advisory Committee, a group of seven scientists convened by the US government in 1964. The US government was concerned that there was a lack of progress being made despite significant expenditure. It concluded that machine translation was more expensive, less accurate and slower than human translation, and that despite the expenses, machine translation was not likely to reach the quality of a human translator in the near future.

The report, however, recommended that tools be developed to aid translators — automatic dictionaries, for example — and that some research in computational linguistics should continue to be supported.

The publication of the report had a profound impact on research into machine translation in the United States, and to a lesser extent the Soviet Union and United Kingdom. Research, at least in the US, was almost completely abandoned for over a decade. In Canada, France and Germany, however, research continued. In the US the main exceptions were the founders of Systran (Peter Toma) and Logos (Bernard Scott), who established their companies in 1968 and 1970 respectively and served the US Dept of Defense. In 1970, the Systran system was installed for the United States Air Force and subsequently in 1976 by the Commission of the European Communities. The METEO System, developed at the Université de Montréal, was installed in Canada in 1977 to translate weather forecasts from English to French, and was translating close to 80,000 words per day or 30 million words per year until it was replaced by a competitor's system on the 30th September, 2001.

While research in the 1960s concentrated on limited language pairs and input, demand in the 1970s was for low-cost systems that could translate a range of technical and commercial documents. This demand was spurred by the increase of globalisation and the demand for translation in Canada, Europe, and Japan.

The 1980s and early 1990s

By the 1980s, both the diversity and the number of installed systems for machine translation had increased. A number of systems relying on mainframe technology were in use, such as Systran, Logos, and Metal.

As a result of the improved availability of microcomputers, there was a market for lower-end machine translation systems. Many companies took advantage of this in Europe, Japan, and the USA. Systems were also brought onto the market in China, Eastern Europe, Korea, and the Soviet Union.

During the 1980s there was a lot of activity in MT in Japan especially. With the Fifth generation computer Japan intended to leap over its competition in computer hardware and software, and one project that many large Japanese electronics firms found themselves involved in was creating software for translating to and from English (Fujitsu, Toshiba, NTT, Brother, Catena, Matsushita, Mitsubishi, Sharp, Sanyo, Hitachi, NEC, Panasonic, Kodensha, Nova, Oki).

Research during the 1980s typically relied on translation through some variety of intermediary linguistic representation involving morphological, syntactic, and semantic analysis.

At the end of the 1980s there was a large surge in a number of novel methods for machine translation. One system was developed at IBM that was based on statistical

methods. Makoto Nagao and his group used methods based on large numbers of example translations, a technique which is now termed example-based machine translation. A defining feature of both of these approaches was the lack of syntactic and semantic rules and reliance instead on the manipulation of large text corpora.

During the 1990s, encouraged by successes in speech recognition and speech synthesis, research began into speech translation with the development of the German Verbmobil project.

There was significant growth in the use of machine translation as a result of the advent of low-cost and more powerful computers. It was in the early 1990s that machine translation began to make the transition away from large mainframe computers toward personal computers and workstations. Two companies that led the PC market for a time were Globalink and MicroTac, following which a merger of the two companies (in December 1994) was found to be in the corporate interest of both. Intergraph and Systran also began to offer PC versions around this time. Sites also became available on the internet, such as AltaVista's Babel Fish (using Systran technology) and Google Language Tools (also initially using Systran technology exclusively).

Recent research

The field of machine translation has in the last few years seen major changes. Currently a large amount of research is being done into statistical machine translation and example-based machine translation. In the area of speech translation, research has focused on moving from domain-limited systems to domain-unlimited translation systems. In different research projects in Europe (like) and in the United States (STR-DUST and) solutions for automatically translating Parliamentary speeches and broadcast news have been developed. In these scenarios the domain of the content is no longer limited to any special area, but rather the speeches to be translated cover a variety of topics. More recently, the French-German project Quaero investigates possibilities to make use of machine translations for a multi-lingual internet. The project seeks to translate not only webpages, but also videos and audio files found on the internet.

Today, only a few companies use statistical machine translation commercially, e.g. Language Weaver (sells translation products and services), Google (uses their proprietary statistical MT system for some language combinations in Google's language tools), Microsoft (uses their proprietary statistical MT system to translate knowledge base articles), and Ta with you (offers a domain-adapted machine translation solution based on statistical MT with some linguistic knowledge). There has been a renewed interest in hybridisation, with researchers combining syntactic and morphological (i.e., linguistic) knowledge into statistical systems, as well as combining statistics with existing rule-based systems.

Translation process

The translation process may be stated as:

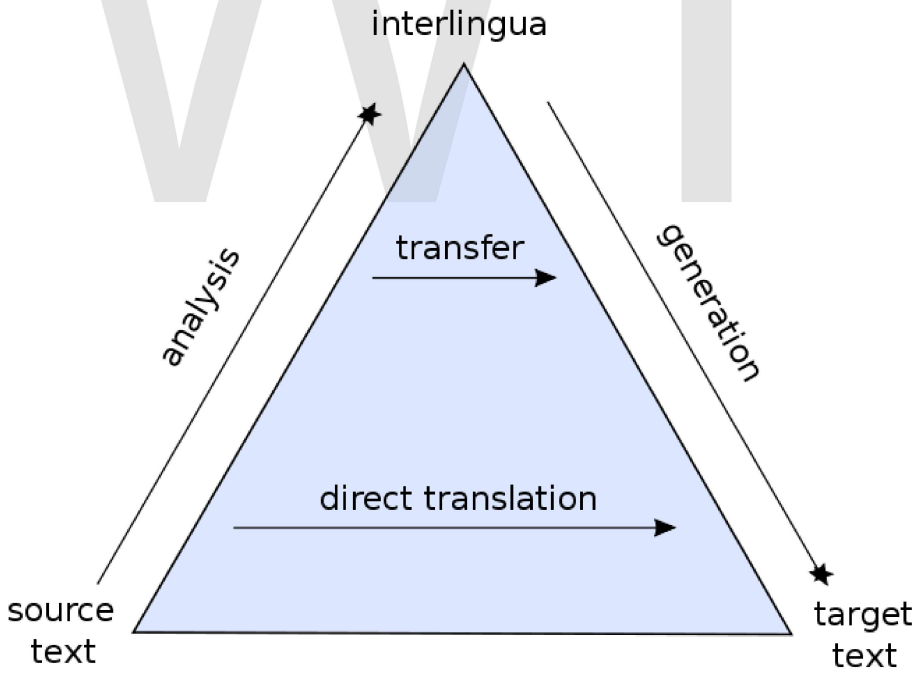
1. Decoding the meaning of the source text; and
2. Re-encoding this meaning in the target language.

Behind this ostensibly simple procedure lies a complex cognitive operation. To decode the meaning of the source text in its entirety, the translator must interpret and analyse all the features of the text, a process that requires in-depth knowledge of the grammar, semantics, syntax, idioms, etc., of the source language, as well as the culture of its speakers. The translator needs the same in-depth knowledge to re-encode the meaning in the target language.

Therein lies the challenge in machine translation: how to program a computer that will "understand" a text as a person does, and that will "create" a new text in the target language that "sounds" as if it has been written by a person.

This problem may be approached in a number of ways.

Approaches



Pyramid showing comparative depths of intermediary representation, interlingual machine translation at the peak, followed by transfer-based, then direct translation.

Machine translation can use a method based on linguistic rules, which means that words will be translated in a linguistic way — the most suitable (orally speaking) words of the target language will replace the ones in the source language.

It is often argued that the success of machine translation requires the problem of natural language understanding to be solved first.

Generally, rule-based methods parse a text, usually creating an intermediary, symbolic representation, from which the text in the target language is generated. According to the nature of the intermediary representation, an approach is described as interlingual machine translation or transfer-based machine translation. These methods require extensive lexicons with morphological, syntactic, and semantic information, and large sets of rules.

Given enough data, machine translation programs often work well enough for a native speaker of one language to get the approximate meaning of what is written by the other native speaker. The difficulty is getting enough data of the right kind to support the particular method. For example, the large multilingual corpus of data needed for statistical methods to work is not necessary for the grammar-based methods. But then, the grammar methods need a skilled linguist to carefully design the grammar that they use.

To translate between closely related languages, a technique referred to as shallow-transfer machine translation may be used.

Rule-based

The rule-based machine translation paradigm includes transfer-based machine translation, interlingual machine translation and dictionary-based machine translation paradigms.

Transfer-based machine translation

Transfer-based machine translation is a type of machine translation. It is based on the idea of interlingua and is currently one of the most widely used methods of machine translation

Overview

Both transfer-based and interlingua-based machine translation have the same idea: to make a translation it is necessary to have an intermediate representation that captures the "meaning" of the original sentence in order to generate the correct translation. In interlingua-based MT this intermediate representation must be independent of the languages in question, whereas in transfer-based MT, it has some dependence on the language pair involved.

The way in which transfer-based machine translation systems work varies substantially, but in general they follow the same pattern: they apply sets of linguistic rules which are

defined as correspondences between the structure of the source language and that of the target language. The first stage involves analysing the input text for morphology and syntax (and sometimes semantics) to create an internal representation. The translation is generated from this representation using both bilingual dictionaries and grammatical rules.

It is possible with this translation strategy to obtain fairly high quality translations, with accuracy in the region of 90% (although this is highly dependent on the language pair in question — for example the distance between the two).

How it works

In a rule-based machine translation system the original text is first analysed morphologically and syntactically in order to obtain a syntactic representation. This representation can then be refined to a more abstract level putting emphasis on the parts relevant for translation and ignoring other types of information. The transfer process then converts this final representation (still in the original language) to a representation of the same level of abstraction in the target language. These two representations are referred to as "intermediate" representations. From the target language representation, the stages are then applied in reverse.

Analysis and transformation

Various methods of analysis and transformation can be used before obtaining the final result. Along with these statistical approaches may be augmented generating hybrid systems. The methods which are chosen and the emphasis depends largely on the design of the system, however, most systems include at least the following stages:

- **Morphological analysis**. Surface forms of the input text are classified as to part-of-speech (e.g. noun, verb, etc.) and sub-category (number, gender, tense, etc.) All of the possible "analyses" for each surface form are typically outputted at this stage, along with the lemma of the word.
- **Lexical categorisation**. In any given text some of the words may have more than one meaning, causing ambiguity in analysis. Lexical categorisation looks at the context of a word to try and determine the correct meaning in the context of the input. This can involve part-of-speech tagging and word sense disambiguation.
- **Lexical transfer**. This is basically dictionary translation, the source language lemma (perhaps with sense information) is looked up in a bilingual dictionary and the translation is chosen.
- **Structural transfer**. While the previous stages deal with words, this stage deals with larger constituents, for example phrases and chunks. Typical features of this stage include concordance of gender and number, and re-ordering of words or phrases.
- **Morphological generation**. From the output of the structural transfer stage, the target language surface forms are generated.

Transfer types

One of the main features of transfer based machine translation systems is a phase that "transfers" an intermediate representation of the text in the original language to an intermediate representation of text in the target language. This can work at one of two levels of linguistic analysis, or somewhere in between. The levels are:

- **Superficial transfer (or syntactic)**. This level is characterised by transferring "syntactic structures" between the source and target languages. It is suitable for languages in the same family or of the same type, for example in the Romance languages between Spanish, Catalan, French, Italian, etc.
- **Deep transfer (or semantic)**. This level constructs a semantic representation that is dependent on the source language. This representation can consist of a series of structures which represent the meaning. In these transfer systems predicates are typically produced. The translation also typically requires structural transfer. This level is used to translate between more distantly related languages (e.g. Spanish-English or Spanish-Basque, etc.)

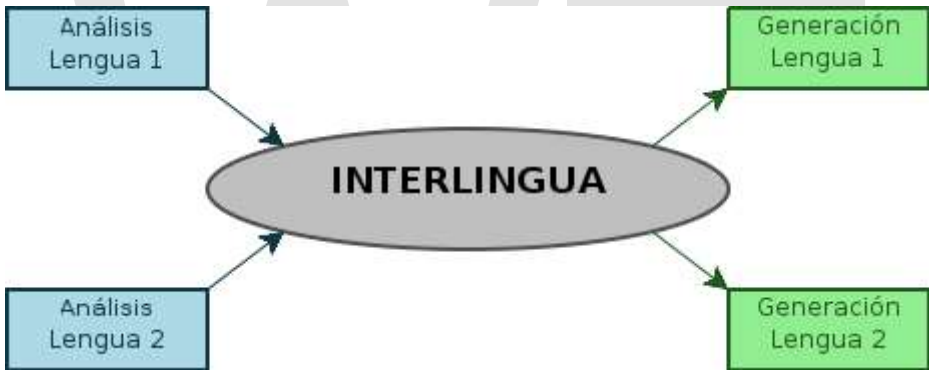


Figure 1. Demonstration of the languages which are used in the process of translating using a *bridge* language.

Interlingual machine translation is one of the classic approaches to machine translation. In this approach, the source language, i.e. the text to be translated is transformed into an interlingua, i.e., an abstract language-independent representation. The target language is then generated from the interlingua. Within the rule-based machine translation paradigm, the interlingual approach is an alternative to the direct approach and the transfer approach.

In the direct approach, words are translated directly without passing through an additional representation. In the transfer approach the source language is transformed into an abstract, less language-specific representation. Linguistic rules which are specific to the language pair then transform the source language representation into an abstract target language representation and from this the target sentence is generated.

The interlingual approach to machine translation has advantages and disadvantages. The advantages are that it requires fewer components in order to relate each source language to each target language, it takes fewer components to add a new language, it supports paraphrases of the input in the original language, it allows both the analyzers and generators to be written by monolingual system developers, and it handles languages that are very different from each other (e.g. English and Arabic). The obvious disadvantage is that the definition of an interlingua is difficult and maybe even impossible for a wider domain. The ideal context for interlingual machine translation is thus multilingual machine translation in a very specific domain.

History

The first ideas about interlingual machine translation appeared in the 17th century with Descartes and Leibniz, who came up with theories of how to create dictionaries using universal numerical codes. Others, such as Cave Beck, Athanasius Kircher and Johann Joachim Becher worked on developing an unambiguous universal language based on the principles logic and iconographs. In 1668, John Wilkins described his interlingua in his "Essay towards a Real Character and a Philosophical Language". In the 18th and 19th centuries many proposals for "universal" international languages were developed, the most well known being Esperanto.

That said, applying the idea of a universal language to machine translation did not appear in any of the first significant approaches. Instead, work started on pairs of languages. However, during the 1950s and 60s, researchers in Cambridge headed by Margaret Masterman, in Leningrad headed by Nikolai Andrev and in Milan by Silvio Ceccato started work in this area. The idea was discussed extensively by the Israeli philosopher Yehoshua Bar-Hillel in 1969.

During the 1970s, noteworthy research was done in Grenoble by researchers attempting to translate physics and mathematical texts from Russian to French, and in Texas a similar project (METAL) was ongoing for Russian to English. Early interlingual MT systems were also built at Stanford in the 1970s by Roger Schank and Yorick Wilks; the former became the basis of a commercial system for the transfer of funds, and the latter's code is preserved at The Computer Museum at Boston as the first interlingual machine translation system.

In the 1980s, renewed relevance was given to interlingua-based, and knowledge-based approaches to machine translation in general, with much research going on in the field. The uniting factor in this research was that high-quality translation required abandoning the idea of requiring total comprehension of the text. Instead, the translation should be based on linguistic knowledge and the specific domain in which the system would be used. The most important research of this era was done in distributed language translation (DLT) in Utrecht, which worked with a modified version of Esperanto, and the Fujitsu system in Japan.

Outline

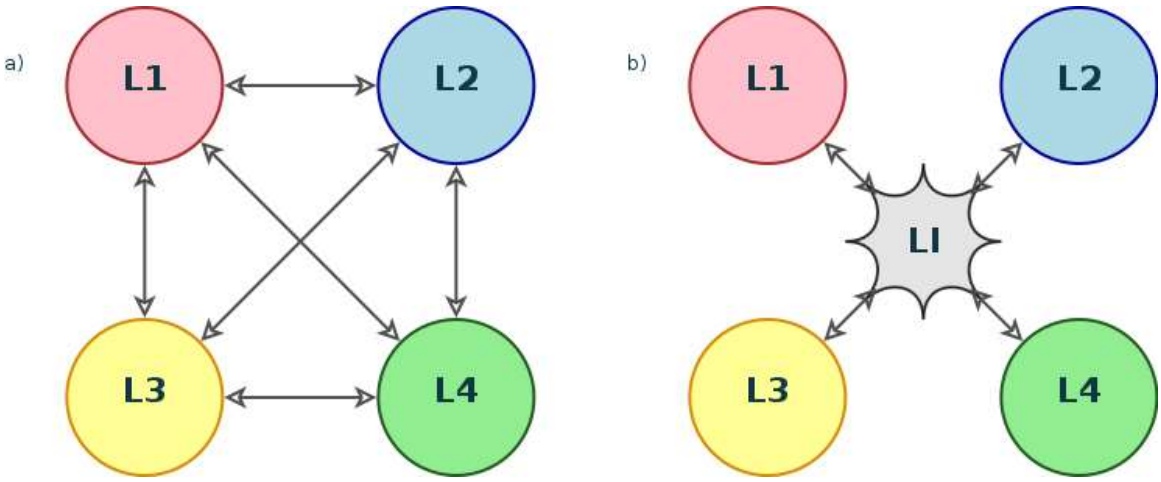


Figure 2. a) Translation graph required for direct or transfer-based machine translation (12 dictionaries are required); b) Translation graph required when using a bridge language (only 8 translation modules are required).

In this method of translation, the interlingua can be thought of as a way of describing the analysis of a text written in a **source language** such that it is possible to convert its morphological, syntactic, semantic (and even pragmatic) characteristics, that is "meaning" into a **target language**. This interlingua is able to describe all of the characteristics of all of the languages which are to be translated, instead of simply translating from one language to another.

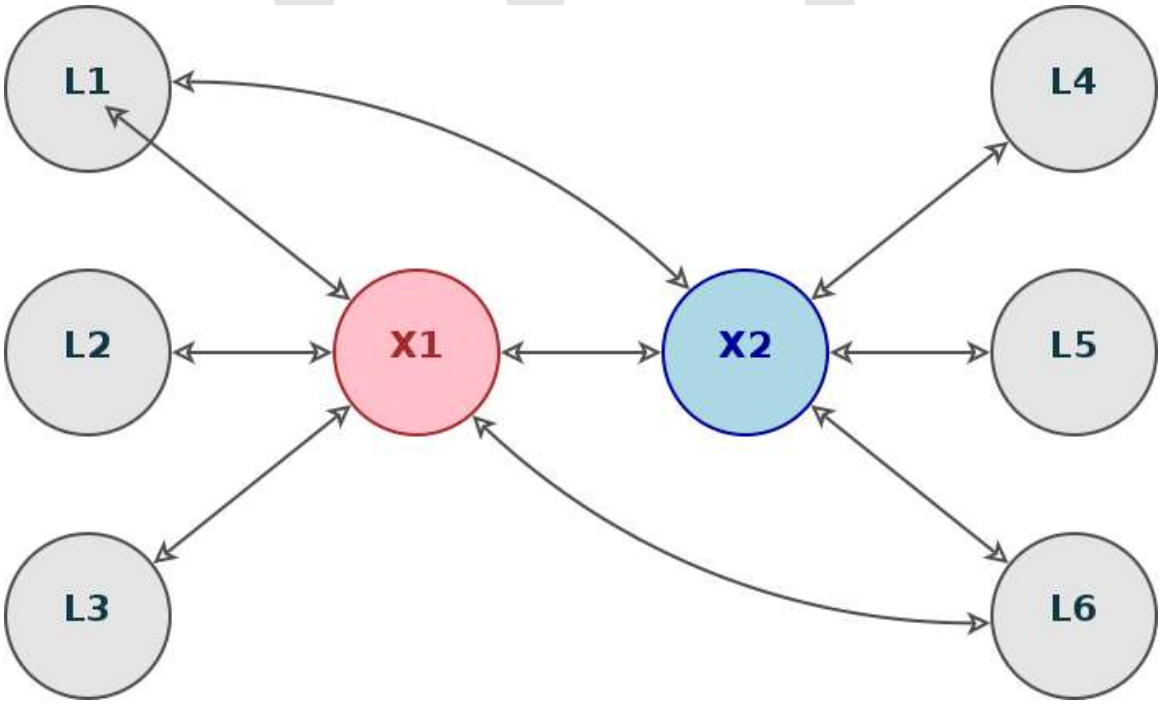


Figure 3: Translation graph using two interlinguas.

Sometimes two interlinguas are used in translation. It is possible that one of the two covers more of the characteristics of the source language, and the other possess more of the characteristics of the target language. The translation then proceeds by converting sentences from the first language into sentences closer to the target language through two stages. The system may also be set up such that the second interlingua uses a more specific vocabulary that is closer, or more aligned with the target language, and this could improve the translation quality.

The above-mentioned system is based on the idea of using linguistic proximity to improve the translation quality from a text in one original language to many other structurally similar languages from only one original analysis. This principle is also used in pivot machine translation, where a natural language is used as a "bridge" between two more distant languages. For example in the case of translating to English from Ukrainian using Russian as an intermediate language.

Translation process

In interlingual machine translation systems, there are two monolingual components: the *analysis* of the source language and the interlingual, and the *generation* of the interlingua and the target language. It is however necessary to distinguish between interlingual systems using only syntactic methods (for example the systems developed in the 1970s at the universities of Grenoble and Texas) and those based on artificial intelligence (from 1987 in Japan and the research at the universities of Southern California and Carnegie Mellon). The first type of system corresponds to that outlined in Figure 1. while the other types would be approximated by the diagram in Figure 4.

The following resources are necessary to an interlingual machine translation system:

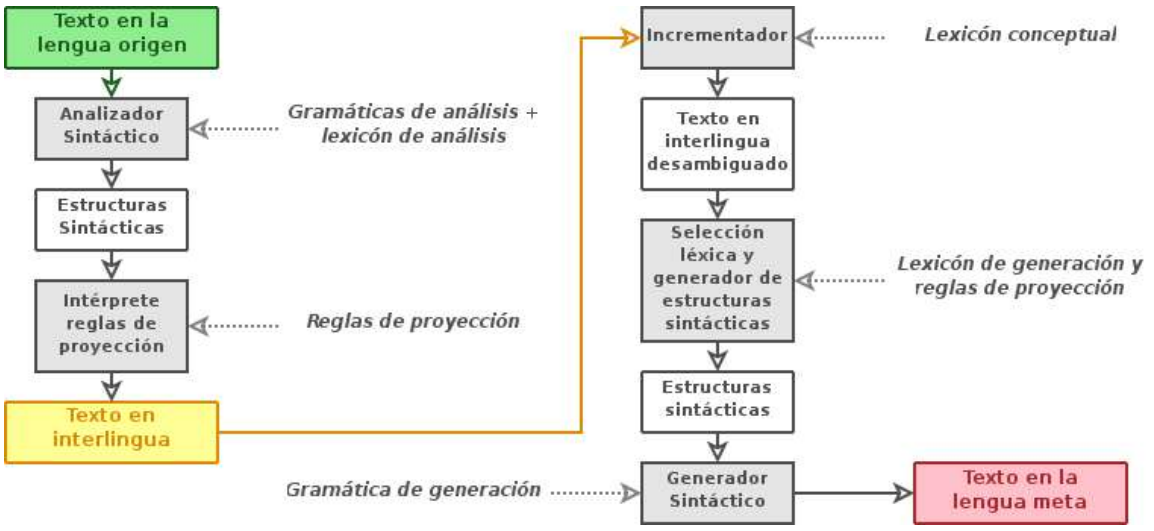


Figure 4. Machine translation in a knowledge-based system.

- Dictionaries (or lexicons) for analysis and generation (specific to the domain and the languages involved).
- A conceptual lexicon (specific to the domain), which is the knowledge base about events and entities known in the domain.
- A set of projection rules (specific to the domain and the languages).
- Grammars for the analysis and generation of the languages involved.

One of the problems of knowledge-based machine translation systems is that it becomes impossible to create databases for domains larger than very specific areas. Another is that processing these databases is very computationally expensive.

Efficacy

One of the main advantages of this strategy is that it provides an economical way to make multilingual translation systems. With an interlingua it becomes unnecessary to make a translation pair between each pair of languages in the system. So instead of creating $n(n-1)$ language pairs, where n is the number of languages in the system, it is only necessary to make $2n$ pairs between the n languages and the interlingua.

The main disadvantage of this strategy is the difficulty of creating an adequate interlingua. It should be both abstract and independent of the source and target languages. The more languages added to the translation system, and the more different they are, the more potent the interlingua must be to express all possible translation directions. Another problem is that it is difficult to extract meaning from texts in the original languages to create the intermediate representation.

Dictionary-based

Machine translation can use a method based on dictionary entries, which means that the words will be translated as they are by a dictionary.

Statistical

Statistical machine translation (**SMT**) is a machine translation paradigm where translations are generated on the basis of statistical models whose parameters are derived from the analysis of bilingual text corpora. The statistical approach contrasts with the rule-based approaches to machine translation as well as with example-based machine translation.

The first ideas of statistical machine translation were introduced by Warren Weaver in 1949, including the ideas of applying Claude Shannon's information theory. Statistical machine translation was re-introduced in 1991 by researchers at IBM's Thomas J. Watson Research Center and has contributed to the significant resurgence in interest in machine

translation in recent years. Nowadays it is by far the most widely-studied machine translation method.

Basis

The idea behind statistical machine translation comes from information theory. A document is translated according to the probability distribution $p(e \mid f)$ that a string e in the target language (for example, English) is the translation of a string f in the source language (for example, French).

The problem of modeling the probability distribution $p(e \mid f)$ has been approached in a number of ways. One intuitive approach is to apply Bayes Theorem, that is $p(e|f) \propto p(f|e)p(e)$, where the translation model $p(f \mid e)$ is the probability that the source string is the translation of the target string, and the language model $p(e)$ is the probability of seeing that target language string. This decomposition is attractive as it splits the problem into two subproblems. Finding the best translation $\tilde{e}$ is done by picking up the one that gives the highest probability:

$$\tilde{e} = arg \max_{e \in e^*} p(e|f) = arg \max_{e \in e^*} p(f|e)p(e)$$

For a rigorous implementation of this one would have to perform an exhaustive search by going through all strings e^* in the native language. Performing the search efficiently is the work of a machine translation decoder that uses the foreign string, heuristics and other methods to limit the search space and at the same time keeping acceptable quality. This trade-off between quality and time usage can also be found in speech recognition.

As the translation systems are not able to store all native strings and their translations, a document is typically translated sentence by sentence, but even this is not enough. Language models are typically approximated by smoothed n-gram models, and similar approaches have been applied to translation models, but there is additional complexity due to different sentence lengths and word orders in the languages.

The statistical translation models were initially word based (Models 1-5 from IBM Hidden Markov model from Stephan Vogel and Model 6 from Franz-Joseph Och), but significant advances were made with the introduction of phrase based models. Recent work has incorporated syntax or quasi-syntactic structures.

Benefits

The most frequently cited benefits of statistical machine translation over traditional paradigms are:

- **Better use of resources**
  - There is a great deal of natural language in machine-readable format.

    - Generally, SMT systems are not tailored to any specific pair of languages.
    - Rule-based translation systems require the manual development of linguistic rules, which can be costly, and which often do not generalize to other languages.
- **More natural translations**

Word-based translation

In word-based translation, the fundamental unit of translation is a word in some natural language. Typically, the number of words in translated sentences are different, because of compound words, morphology and idioms. The ratio of the lengths of sequences of translated words is called fertility, which tells how many foreign words each native word produces. Necessarily it is assumed by information theory that each covers the same concept. In practice this is not really true. For example, the English word *corner* can be translated in Spanish by either *rincón* or *esquina*, depending on whether it is to mean its internal or external angle.

Simple word-based translation can't translate between languages with different fertility. Word-based translation systems can relatively simply be made to cope with high fertility, but they could map a single word to multiple words, but not the other way about. For example, if we were translating from French to English, each word in English could produce any number of French words— sometimes none at all. But there's no way to group two English words producing a single French word.

An example of a word-based translation system is the freely available GIZA++ package (GPLed), which includes the training program for IBM models and HMM model and Model 6.

The word-based translation is not widely used today; phrase-based systems are more common. Most phrase-based system are still using GIZA++ to align the corpus. The alignments are used to extract phrases or deduce syntax rules. And matching words in bi-text is still a problem actively discussed in the community. Because of the predominance of GIZA++, there are now several distributed implementations of it online.

Phrase-based translation

In phrase-based translation, the aim is to reduce the restrictions of word-based translation by translating whole sequences of words, where the lengths may differ. The sequences of words are called blocks or phrases, but typically are not linguistic phrases but phrases found using statistical methods from corpora. It has been shown that restricting the phrases to linguistic phrases decreases the quality of translation

Syntax-based translation

Syntax-based translation is based on the idea of translating syntactic units, rather than single words or strings of words (as in phrase-based MT), i.e. (partial) parse trees of sentences/utterances. The idea of syntax-based translation is quite old in MT, though it's statistical counterpart did not take off until the advent of strong stochastic parsers in the 1990s. Examples of this approach include DOP-based MT and, more recently, synchronous context-free grammars.

Challenges with statistical machine translation

Problems that statistical machine translation have to deal with include

Sentence alignment

In parallel corpora single sentences in one language can be found translated into several sentences in the other and vice versa. Sentence aligning can be performed through the Gale-Church alignment algorithm.

Compound words

Idioms

Depending on the corpora used, idioms may not translate "idiomatically". For example, using Canadian Hansard as the bilingual corpus, "hear" may almost invariably be translated to "Bravo!" since in Parliament "Hear, Hear!" becomes "Bravo!".

Morphology

Different word orders

Word order in languages differ. Some classification can be done by naming the typical order of subject (S), verb (V) and object (O) in a sentence and one can talk, for instance, of SVO or VSO languages. There are also additional differences in word orders, for instance, where modifiers for nouns are located, or where the same words are used as a question or a statement.

In speech recognition, the speech signal and the corresponding textual representation can be mapped to each other in blocks in order. This is not always the case with the same text in two languages. For SMT, the machine translator can only manage small sequences of words, and word order has to be thought of by the program designer. Attempts at solutions have included re-ordering models, where a distribution of location changes for each item of translation is guessed from aligned bi-text. Different location changes can be ranked with the help of the language model and the best can be selected.

Example-based

The **example-based machine translation** (**EBMT**) approach to machine translation is often characterized by its use of a bilingual corpus with *parallel texts* as its main knowledge base, at run-time. It is essentially a translation by analogy and can be viewed as an implementation of case-based reasoning approach of machine learning.

At the foundation of example-based machine translation is the idea of translation by analogy. When applied to the process of human translation, the idea that translation takes place by analogy is a rejection of the idea that people translate sentences by doing deep linguistic analysis. Instead it is founded on the belief that people translate firstly by decomposing a sentence into certain phrases, then by translating these phrases, and finally by properly composing these fragments into one long sentence. Phrasal translations are translated by analogy to previous translations. The principle of translation by analogy is encoded to example-based machine translation through the example translations that are used to train such a system.

Example of bilingual corpus

English	Japanese
How much is that **red umbrella**?	Ano **akai kasa** wa ikura desu ka.
How much is that **small camera**?	Ano **chiisai kamera** wa ikura desu ka.

Example-based machine translation systems are trained from bilingual parallel corpora, which contain sentence pairs like the example shown in the table. Sentence pairs contain sentences in one language with their translations into another. The particular example shows an example of a *minimal pair*, meaning that the sentences vary by just one element. These sentences make it simple to learn translations of subsentential units. For example, an example-based machine translation system would learn three units of translation:

1. *How much is that* **X** *?* corresponds to *Ano **X** wa ikura desu ka.*
2. *red umbrella* corresponds to *akai kasa*
3. *small camera* corresponds to *chiisai kamera*

Composing these units can be used to produce novel translations in the future. For example, if we have been trained using some text containing the sentences:

President Kennedy was shot dead during the parade. and *The convict escaped on July 15th.* We could translate the sentence *The convict was shot dead during the parade.* by substituting the appropriate parts of the sentences.

Other approaches to machine translation, including statistical machine translation, also use bilingual corpora to learn the process of translation.

Example based machine translation was first suggested by Makoto Nagao in 1984. It soon attracted the attention of scientists in the field of natural language processing.

EBMT is best suited for sub-language phenomena like phrasal verbs.

Phrasal verbs have highly context-dependent meanings. Phrasal verbs are a commonly occurring feature in English and comprise a verb followed by an adverb and/or a preposition. The adverb/preposition(s) are termed as the particle to the verb. Phrasal verbs produce specialized context-specific meanings that may not be derived from the meaning of the constituents. There is almost always an ambiguity during word-to-word translation from source to the target language.

As an example, let us consider the phrasal verb: put on and its Hindi meaning. It may be used in any of the following ways: Ram put on the lights. (Switched on) (Jalana) Ram put on a cap. (Wear) (Pahenna)

EBMT can be used to determine the context of the sentence.

Hybrid MT

Hybrid machine translation (HMT) leverages the strengths of statistical and rule-based translation methodologies. Several MT companies (Asia Online, LinguaSys, and Systran) are claiming to have a hybrid approach using both rules and statistics. The approaches differ in a number of ways:

- **Rules post-processed by statistics**: Translations are performed using a rules based engine. Statistics are then used in an attempt to adjust/correct the output from the rules engine.
- **Statistics guided by rules**: Rules are used to pre-process data in an attempt to better guide the statistical engine. Rules are also used to post-process the statistical output to perform functions such as normalization. This approach has a lot more power, flexibility and control when translating.

Major issues

Disambiguation

Word-sense disambiguation concerns finding a suitable translation when a word can have more than one meaning. The problem was first raised in the 1950s by Yehoshua Bar-Hillel. He pointed out that without a "universal encyclopedia", a machine would never be able to distinguish between the two meanings of a word. Today there are numerous approaches designed to overcome this problem. They can be approximately divided into "shallow" approaches and "deep" approaches.

Shallow approaches assume no knowledge of the text. They simply apply statistical methods to the words surrounding the ambiguous word. Deep approaches presume a

comprehensive knowledge of the word. So far, shallow approaches have been more successful.

The late Claude Piron, a long-time translator for the United Nations and the World Health Organization, wrote that machine translation, at its best, automates the easier part of a translator's job; the harder and more time-consuming part usually involves doing extensive research to resolve ambiguities in the source text, which the grammatical and lexical exigencies of the target language require to be resolved:

> Why does a translator need a whole workday to translate five pages, and not an hour or two? ..... About 90% of an average text corresponds to these simple conditions. But unfortunately, there's the other 10%. It's that part that requires six [more] hours of work. There are ambiguities one has to resolve. For instance, the author of the source text, an Australian physician, cited the example of an epidemic which was declared during World War II in a "Japanese prisoner of war camp". Was he talking about an American camp with Japanese prisoners or a Japanese camp with American prisoners? The English has two senses. It's necessary therefore to do research, maybe to the extent of a phone call to Australia.

The ideal deep approach would require the translation software to do all the research necessary for this kind of disambiguation on its own; but this would require a higher degree of AI than has yet been attained. A shallow approach which simply guessed at the sense of the ambiguous English phrase that Piron mentions (based, perhaps, on which kind of prisoner-of-war camp is more often mentioned in a given corpus) would have a reasonable chance of guessing wrong fairly often. A shallow approach that involves "ask the user about each ambiguity" would, by Piron's estimate, only automate about 25% of a professional translator's job, leaving the harder 75% still to be done by a human.

Named entities

Related to named entity recognition in information extraction.

Applications

There are now many software programs for translating natural language, several of them online, such as:

- LinguaSys provides highly customized hybrid machine translation that can go from any language to any language.
- Asia Online provides a custom machine translation engine building capability that they claim gives near-human quality compared to the "gist" based quality of free online engines. Asia Online also provides tools to edit and create custom machine translation engines with their Language Studio suite of products.

- Hindi to Punjabi Machine Translation System, provides machine translation using a direct approach. It translates Hindi into Punjabi. It also features writing e-mail in the Hindi language and sending the same in Punjabi to the recipient.
- Arabic machine translation in multilingual framework.
- Worldlingo provides machine translation using both statistical based TE's and rule based TE's. Most recognizable as the MT partner in Microsoft Windows and Microsoft Mac Office.
- Power Translator
- SYSTRAN, which powers Yahoo! Babel Fish
- Promt, which powers online translation services at Voila.fr and Orange.fr
- AppTek, which released a hybrid MT system in 2009.
- Toggletext uses a transfer-based system (known as Kataku) to translate between English and Indonesian.
- Anusaaraka A free open source machine translation from English to Hindi based on Panini grammar and uses state of the art NLP tools. Can be used online and downloaded from
- Apertium, a free and open source machine translation platform (WinXLator gives this a Windows GUI, but it is likely to be in violation of the Apertium GPL license)
- Google Translator A free online translator from Google.

Other translation software, most of them running under Microsoft Windows, includes

- Translation memory tools, such as SDL Trados, Wordfast, Deja Vu, Swordfish, and
- localization tools, such and Alchemy CATALYST and Multilizer.

(A comparison test of software of this kind may be seen here.) A number of translation software programs are available free of charge, e.g. ForeignDesk and the multiplatform Okapi Framework and OmegaT+.

While no system provides the holy grail of fully-automatic high-quality machine translation of unrestricted text, many fully-automated systems produce reasonable output. The quality of machine translation is substantially improved if the domain is restricted and controlled.

Despite their inherent limitations, MT programs are used around the world. Probably the largest institutional user is the European Commission. The MOLTO project, for example, coordinated by the University of Gothenburg, received more than 2.375 million euros project support from the EU to create a reliable translation tool that covers a majority of the EU languages.

Google has claimed that promising results were obtained using a proprietary statistical machine translation engine. The statistical translation engine used in the Google language tools for Arabic <-> English and Chinese <-> English had an overall score of 0.4281 over

the runner-up IBM's BLEU-4 score of 0.3954 (Summer 2006) in tests conducted by the National Institute for Standards and Technology.

With the recent focus on terrorism, the military sources in the United States have been investing significant amounts of money in natural language engineering. *In-Q-Tel* (a venture capital fund, largely funded by the US Intelligence Community, to stimulate new technologies through private sector entrepreneurs) brought up companies like Language Weaver. Currently the military community is interested in translation and processing of languages like Arabic, Pashto, and Dari. The Information Processing Technology Office in DARPA hosts programs like TIDES and Babylon Translator. US Air Force has awarded a $1 million contract to develop a language translation technology.

The notable rise of social networking on the web in recent years has created yet another niche for the application of machine translation software – in utilities such as Facebook, or instant messaging clients such as Skype, GoogleTalk, MSN Messenger, etc. – allowing users speaking different languages to communicate with each other. Machine translation applications have also been released for most mobile devices, including mobile telephones, pocket PCs, PDAs, etc. Due to their portability, such instruments have come to be designated as mobile translation tools enabling mobile business networking between partners speaking different languages, or facilitating both foreign language learning and unaccompanied traveling to foreign countries without the need of the intermediation of a human translator.

Evaluation

Machine translation systems and output can be evaluated along numerous dimensions. The intended use of the translation, characteristics of the MT software, the nature of the translation process, etc., all affect how one evaluates MT systems and their output.

There are various means for evaluating the output quality of machine translation systems. The oldest is the use of human judges to assess a translation's quality. Even though human evaluation is time-consuming, it is still the most reliable way to compare different systems such as rule-based and statistical systems. Automated means of evaluation include BLEU, NIST and METEOR.

Relying exclusively on unedited machine translation ignores the fact that communication in human language is context-embedded and that it takes a person to comprehend the context of the original text with a reasonable degree of probability. It is certainly true that even purely human-generated translations are prone to error. Therefore, to ensure that a machine-generated translation will be useful to a human being and that publishable-quality translation is achieved, such translations must be reviewed and edited by a human. The late Claude Piron wrote that machine translation, at its best, automates the easier part of a translator's job; the harder and more time-consuming part usually involves doing extensive research to resolve ambiguities in the source text, which the grammatical and lexical exigencies of the target language require to be resolved. Such research is a

necessary prelude to the pre-editing necessary in order to provide input for machine-translation software such that the output will not be meaningless.

In certain applications, however, e.g., product descriptions written in a controlled language, a dictionary-based machine-translation system has produced satisfactory translations that require no human intervention save for quality inspection.

Chapter- 4

Telematics

Telematics typically is any integrated use of telecommunications and informatics, also known as ICT (Information and Communications Technology). Hence the application of telematics is with any of the following:

Lexus Gen V navigation system

- The technology of sending, receiving and storing information via telecommunication devices in conjunction with effecting control on remote objects.
- The integrated use of telecommunications and informatics, for application in vehicles and with control of vehicles on the move.
- Telematics includes but is not limited to Global Positioning System technology integrated with computers and mobile communications technology in automotive navigation systems.
- Most narrowly, the term has evolved to refer to the use of such systems within road vehicles, in which case the term **vehicle telematics** may be used.

In contrast *telemetry* is the transmission of measurements from the location of origin to the location of computing and consumption, especially without effecting control on the remote objects. Telemetry is typically applied in testing of flight objects.

Although the majority of devices hat integrate telecommunications and information technology are not vehicles but rather mobile phones and the like, their use is not included in telematics.

Vehicle telematics

The etymology of *telematics*, as determined by Automotive Telematics author and academic Dennis Foy, is from the Greek "tele" ('far away', especially in relation to the process of producing or recording) and ~Matos (a derivative of the Greek machinari, or contrivance, usually taken in this context to mean 'of its own accord'). As combined, the term "telematics" describes the process of long-distance transmission of computer-based information. It was first introduced in French by Simon Nora and Alain Minc in *L'informatisation de la Société* (La Documentation Française, 1978)

Telematics — 1. The convergence of telecommunications and information processing, the term later evolved to refer to automation in automobiles, such as the invention of the emergency warning system for vehicles. GPS navigation, integrated hands-free cell phones, wireless safety communications and automatic driving assistance systems all are covered under the telematics umbrella. 2. The science of **Tele**communications and Infor**matics** applied in wireless technologies and computational systems. 802.11p, the IEEE standard in the 802.11 family and also referred to as Wireless Access for the Vehicular Environment (WAVE), is the primary standard that addresses and enhances Intelligent Transportation System. 3. Emad Isaac, CTO of the Morey Corporation defines Telematics as "The potential for collection, aggregation, and storage of pertinent data that can be digested locally, or post-processed remotely." While it is applicable to the vehicle market, this definition suggests a more universally applicable technology as a superset of M2M (Machine to Machine) connectivity, and as part of an "intelligent network of connected things".

Practical applications of vehicle telematics

When used in a commercial environment vehicle telematics can potentially be a powerful and valuable tool to improve the efficiency of an organization. Some practical applications of vehicle telematics include;

Telematics education

A project entitled the European Automotive Digital Innovation Studio (EADIS) has been awarded 400,000 Euros from the European commission under its Leonardo programme. EADIS will use a virtual work environment called the Digital Innovation Studio to train and develop professional designers in the automotive industry in the impact and application of ‘vehicle telematics’ so that they may integrate new technologies into future products within the automotive industry.

Leonardo da Vinci is a European Community programme which aims to support national training strategies through funding a range of transnational partnership projects aimed at improving quality, fostering innovation and promoting the European dimension in vocational training. The programme promotes transnational projects based on co-operation between the various players in vocational training - training bodies, vocational

schools, universities, businesses, chambers of commerce, etc. - in an effort to increase mobility, to foster innovation and to improve the quality of training. The Leonardo da Vinci programme aims at helping people improve their skills throughout their lives.

“The European automotive industry is losing competitiveness as challengers from lower-cost economies have increased their share of world automotive markets” (CLEPA, European Association of automotive supplier’s White paper 2005). As a European solution to this problem, EADIS will develop training and infrastructure to enable European companies to operate more innovatively and efficiently.

This project is executed in partnership with:

- Coventry University (CEPAD), UK
- Oulu University of Applied Sciences, Finland
- Munster University of Applied Sciences, Germany
- Turin Polytechnic, Italy
- Technical University of Delft, the Netherlands

An Advisory panel made up of industry representatives including RDM automotive, Ricardo and MIRA has been set up to evaluate the project. All the partners are looking forward to developing the project and using it as a platform for building relationships and collaborating internationally with other universities and industry partners.

Vehicle tracking

Vehicle tracking is a way of monitoring the location, movements, status and behaviour of a vehicle or fleet of vehicles. This is achieved through a combination of a GPS(GNSS) receiver and an electronic device (usually comprising a GSM GPRS modem or SMS sender) installed in each vehicle, communicating with the user (dispatching, emergency or co-ordinating unit) and PC- or web-based software. The data are turned into information by management reporting tools in conjunction with a visual display on computerised mapping software. Vehicle tracking systems may also use odometry or dead reckoning as an alternative or complementary means of navigation.

Trailer tracking

Trailer tracking is the technology of tracking the movements and position of an articulated vehicle's trailer unit, through the use of a location unit fitted to the trailer and a method of returning the position data via mobile communication network or geostationary satellite communications, for use through either PC- or web-based software.

Cold store

Cold store freight trailers that are used to deliver fresh or frozen foods are increasingly incorporating telematics to gather time-series data on the temperature inside the cargo

container, both to trigger alarms and record an audit trail for business purposes. An increasingly sophisticated array of sensors, many incorporating RFID technology, are being used to ensure that temperature throughout the cargo remains within food-safety parameters.

Fleet management

Fleet management is the management of a company's vehicle fleet. Fleet management includes the management of ships and or motor vehicles such as cars, vans and trucks. Fleet (vehicle) Management can include a range of Fleet Management functions, such as vehicle financing, vehicle maintenance, vehicle telematics (tracking and diagnostics), driver management, fuel management and health & safety management. Fleet Management is a function which allows companies which rely on transportation in their business to remove or minimize the risks associated with vehicle investment, improving efficiency, productivity and reducing their overall transportation costs, providing 100% compliancy with government legislation and Duty of Care obligations. These functions can either be dealt with by an in-house Fleet Management department or an outsourced Fleet Management provider.

The Association of Equipment Management Professionals (AEMP) successfully developed the industry's first Telematic Standard.

In 2008, AEMP brought together the major construction equipment manufacturers and telematics providers in the heavy equipment industry to discuss the development of the industry's first telematics standard. Following agreement from Caterpillar, Volvo CE, Komatsu, and John Deere Construction & Forestry to support such a standard, the AEMP formed a standards development subcommittee, chaired by Pat Crail CEM, to develop the standard, . This committee consisted of developers provided by the Caterpillar/Trimble joint venture known as Virtual Site Solutions, Volvo CE, and John Deere. Will McFadyen of McFadyen & Associates provided expertise derived through years of integrating telematics data from various providers into a wide variety of customer fleet management, estimating, and accounting systems. This group worked from February 2009 through September 2010 to develop the industry's first standard for the delivery of telematics data.

> The result, the AEMP Telematics Data Standard V1.1, was released in 2010 and officially went live on October 1st, 2010. As of November 1, 2010, Caterpillar, Volvo CE, John Deere Construction & Forestry, OEM Data Delivery, and Navman Wireless are able to support customers with delivery of basic telematics data in a standard xml format. Komatsu, Topcon, and others are finishing beta testing and have indicated that they will be able to support customers before the end of 2010.
> The AEMP's telematics data standard was developed to allow end users to integrate key telematics data (operating hours, location, fuel consumed, and odometer reading where applicable) into their existing fleet management reporting systems. As such, the standard was primarily intended to facilitate

importation of these data elements into enterprise software systems such as those used by many medium to large construction contractors. Prior to the standard, end users had few options for integrating this data into their reporting systems in a mixed-fleet environment consisting of multiple brands of machines and a mix of telematics-equipped machines and legacy machines (those without telematics devices where operating data is still reported manually via pen and paper). One option available to machine owners was to visit multiple websites to manually retrieve data from each manufacturer's telematics interface and then manually enter it into their fleet management program's database. This option was cumbersome and labor-intensive.

A second option was for the end user to develop an API (Application Programming Interface), or program, to integrate the data from each telematics provider into his or her database. This option was quite costly, as each telematics provider had a different procedure for accessing and retrieving the data and the data format varied from provider to provider. This option automated the process, but because each provider required a unique, custom API to retrieve and parse the data, it was an expensive option. in addition, another API had to be developed any time another brand of machine or telematics device was added to the fleet.

A third option for mixed-fleet integration was to replace the various factory-installed telematics devices with devices from a third party telematics provider. Although this solved the problem of having multiple data providers requiring unique integration methods, this was by far the most expensive option. In addition to the expense, many of the third-party devices available for construction equipment are unable to access data directly from the machine's electronic control modules (ECMs), or computers, and as such are more limited than the device installed by the OEM (Cat, Volvo, Deere, Komatsu, etc) in the data they are able to provide. In some cases, these devices are limited to location and engine run time, although they are increasingly able to accommodate a number of add-on sensors to provide additional data.

The AEMP Telematics Data Standard provides a fourth option. By concentrating on the key data elements that drive the majority of fleet management reports (hours, miles, location, fuel consumption), making those data elements available in a standardized xml format, and standardizing the means by which the document is retrieved, the standard allows the end user to use one API to retrieve data from any participating telematics provider. Because one API can retrieve data from any participating telematics provider, as opposed to the unique API for each provider that was required previously, integration development costs are greatly reduced.

Satellite navigation

Satellite navigation in the context of vehicle telematics is the technology of using a GPS and electronic mapping tool to enable the driver of a vehicle to locate a position, then route plan and navigate a journey.

Mobile data and mobile television

Mobile data is use of wireless data communications using radio waves to send and receive real time computer data to, from and between devices used by field based personnel. These devices can be fitted solely for use while in the vehicle (Fixed Data Terminal) or for use in and out of the vehicle (Mobile Data Terminal).

Mobile data can be used to receive TV channels and programs, in a similar way to mobile phones, but using LCD TV devices.

Wireless vehicle safety communications

Wireless vehicle safety communications telematics aid in car safety and road safety. It is an electronic sub-system in a car or other vehicle for the purpose of exchanging safety information, about such things as road hazards and the locations and speeds of vehicles, over short range radio links. This may involve temporary ad hoc wireless local area networks.

Wireless units will be installed in vehicles and probably also in fixed locations such as near traffic signals and emergency call boxes along the road. Sensors in the cars and at the fixed locations, as well as possible connections to wider networks, will provide the information, which will be displayed to the drivers in some way. The range of the radio links can be extended by forwarding messages along multi-hop paths. Even without fixed units, information about fixed hazards can be maintained by moving vehicles by passing it backwards. It also seems possible for traffic lights, which one can expect to become smarter, to use this information to reduce the chance of collisions.

Further in the future, it may connect directly to the adaptive cruise control or other vehicle control aids. Cars and trucks with the wireless system connected to their brakes may move in convoys, to save fuel and space on the roads. When any column member slows down, all those behind it will automatically slow also. There are also possibilities that need less engineering effort. A radio beacon could be connected to the brake light, for example.

Network ideas are scheduled for test in fall 2008, in Europe where radio frequency bandwidth has been allocated. The 30 MHz allocated is at 5.9 GHz, and unallocated bandwidth at 5.4 GHz may also be used. The standard is IEEE 802.11p, a low latency form of the Wi-Fi local area network standard. Similar efforts are underway in Japan and the USA.

Emergency warning system for vehicles

Telematics technologies are self-orientating open network architecture structure of variable programmable intelligent beacons developed for application in the development of intelligent vehicles — with target intent to accord (blend, or mesh) warning information with surrounding vehicles in the vicinity of travel, intra-vehicle, and infrastructure. Emergency warning system for vehicles telematics particularly developed for international harmonisation and standardisation of vehicle-to-vehicle —

infrastructure-to-vehicle — and vehicle-to-infrastructure real-time Dedicated Short Range Communication (DSRC) systems.

Telematics most commonly relate to computerised systems that update information at the same rate as they receive data, enabling them to direct or control a process such as an instantaneous autonomous warning notification in a remote machine or group of machines. By use of telematics as applied to intelligent vehicle technologies, instantaneous direction travel cognizance of a vehicle may be transmitted in real-time to surrounding vehicles traveling in the local area of vehicles equipped (with EWSV) to receive said warning signals of danger.

Intelligent vehicle technologies

Telematics comprise electronic, electromechanical, and electromagnetic devices — usually silicon micromachined components operating in conjunction with computer controlled devices and radio transceivers to provide precision repeatability functions (such as in robotics artificial intelligence systems) emergency warning validation performance reconstruction.

Intelligent vehicle technologies commonly apply to car safety systems and self-contained autonomous electromechanical sensors generating warnings that can be transmitted within a specified targeted area of interest, say within 100 meters of the emergency warning system for vehicles transceiver. In ground applications, intelligent vehicle technologies are utilized for safety and commercial communications between vehicles or between a vehicle and a sensor along the road.

On November 3, 2009 the most advanced Intelligent Vehicle concept car was demonstrated in New York City. A 2010 Toyota Prius became the first LTE Connected Car. The demonstration was provided by the NG Connect project, a collaboration of automotive telematic technologies designed to exploit in-car 4G wireless network connectivity.

Car clubs

Telematics technology has allowed car clubs to emerge, such as City Car Club in the UK. Telematics-enabled computers allow organisers to track members' usage and bill them on a pay-as-you-drive. Car Clubs such as Australia's Charter Drive use telematics to monitor and report on vehicle use within pre-defined geofence areas, in order to demonstrate the reach of their transit media car club fleet.

Auto insurance

The basic idea of telematic auto insurance is that a driver's behavior is monitored directly while the person drives and this information is transmitted to an insurance company. The insurance company then assesses the risk of that driver having an accident and charges insurance premiums accordingly. A driver who drives long distance at high speed, for

example, will be charged a higher rate than a driver who drives short distances at slower speeds.

Telematic auto insurance was independently invented and patented by a major U.S. auto insurance company, Progressive Auto Insurance U.S. Patent 5,797,134 and a Spanish independent inventor, Salvador Minguijon Perez (European Patent EP0700009B1). The Progressive patents cover the use of a cell phone and GPS to track movements of a car. The Perez patents cover monitoring the car's engine control computer to determine distance driven, speed, time of day, braking force, etc. Ironically, Progressive is developing the Perez technology in the US and European auto insurer Norwich Union is developing the Progressive technology for Europe.

Trials conducted by Norwich Union in 2005 have found that young drivers (18 to 23 year olds) signing up for telematic auto insurance have had a 20% lower accident rate than average.

Recent theoretical economic research on the social welfare effects of Progressive's telematics technology business process patents have questioned whether the business process patents are pareto efficient for society. Preliminary results suggest that it is not, but more work is needed.

Chapter- 5

Home Automation

Home automation (also called **domotics**) is the residential extension of "building automation". It is automation of the home, housework or household activity. Home automation may include centralized control of lighting, HVAC (heating, ventilation and air conditioning), appliances, and other systems, to provide improved convenience, comfort, energy efficiency and security. Home automation for the elderly and disabled can provide increased quality of life for persons who might otherwise require caregivers or institutional care.

A home automation system integrates electrical devices in a house with each other. The techniques employed in home automation include those in building automation as well as the control of domestic activities, such as home entertainment systems, houseplant and yard watering, pet feeding, changing the ambiance "scenes" for different events (such as dinners or parties), and the use of domestic robots. Devices may be connected through a computer network to allow control by a personal computer, and may allow remote access from the internet.

Typically, a new home is outfitted for home automation during construction, due to the accessibility of the walls, outlets, and storage rooms, and the ability to make design changes specifically to accommodate certain technologies. Wireless systems are commonly installed when outfitting a pre-existing house, as they reduce wiring changes. These communicate through the existing power wiring, radio, or infrared signals with a central controller. Network sockets may be installed in every room like AC power receptacles.

Although automated *homes of the future* have been staple exhibits for World's Fairs and popular backgrounds in science fiction, complexity, competition between vendors, multiple incompatible standards and the resulting expense have limited the penetration of home automation to homes of the wealthy or ambitious hobbyists.

A typical domestic patch panel.

Overview and benefits

In modern construction in industrialized nations, homes have been wired for electrical power, telephones, TV outlets (cable or antenna), and a doorbell.

Many household tasks were automated by the development of special appliances. For instance, automatic washing machines were developed to reduce the manual labor of cleaning clothes, and water heaters reduced the labor necessary for bathing.

Other traditional household tasks, like food preservation and preparation have been automated in large extent by moving them into factory settings, with the development of pre-made, pre-packaged foods, and in some countries, such as the United States, increased reliance on commercial food preparation services, such as fast food restaurants. Volume and the factory setting allows forms of automation that would be impractical or too costly in a home setting. Standardized foods enable possible further automation of handling the food within the home.

The use of gaseous or liquid fuels, and later the use of electricity enabled increased automation in heating, reducing the labor necessary to fuel heaters and stoves. Development of thermostats allowed more automated control of heating, and later cooling.

A remote control for moving vessels and vehicles was first patented by Nikola Tesla in 1898.

World's Fairs in Chicago (1934), New York (1939) and (1964–65) depicted electrified and automated homes. In 1966 Jim Sutherland, an engineer working for Westinghouse Electric, developed a home automation system called "ECHO IV"; this was a private project and never commercialized.

With the invention of the microcontroller, the cost of electronic control fell rapidly. Remote and intelligent control technologies were adopted by the building services industry and appliance manufacturers worldwide, as they offer the end user easily accessible and/or greater control of their products.

As the amount of controllable appliances in the home rises, the ability of these devices to interconnect and communicate with each other digitally becomes a useful and desirable feature. The consolidation of control or monitoring signals from appliances, fittings or basic services is an aim of home automation.

In simple installations this may be as straightforward as turning on the lights when a person enters the room. In advanced installations, rooms can sense not only the presence of a person inside but know who that person is and perhaps set appropriate lighting, temperature, music levels or television channels, taking into account the day of the week, the time of day, and other factors.

Other automated tasks may include setting the air conditioning to an energy saving setting when the house is unoccupied, and restoring the normal setting when an occupant is about to return. More sophisticated systems can maintain an inventory of products, recording their usage through bar codes, or an RFID tag, and prepare a shopping list or even automatically order replacements.

Home automation can also provide a remote interface to home appliances or the automation system itself, via telephone line, wireless transmission or the internet, to provide control and monitoring via a smart phone or web browser.

An example of a remote monitoring in home automation could be when a smoke detector detects a fire or smoke condition, then all lights in the house will blink to alert any occupants of the house to the possible fire. If the house is equipped with a home theatre, a home automation system can shut down all audio and video components to avoid distractions, or make an audible announcement. The system could also call the home owner on their mobile phone to alert them, or call the fire department or alarm monitoring company.

System

The elements of a domotics system are:

Programmable logic controller

e

At rack, left-to-right: power supply unit PS407 4A,CPU 416-3, interface module IM 460-0 and communication processor CP 443-1.

A **programmable logic controller** (**PLC**) or **programmable controller** is a digital computer used for automation of electromechanical processes, such as control of machinery on factory assembly lines, amusement rides, or lighting fixtures. PLCs are used in many industries and machines. Unlike general-purpose computers, the PLC is designed for multiple inputs and output arrangements, extended temperature ranges, immunity to electrical noise, and resistance to vibration and impact. Programs to control machine operation are typically stored in battery-backed or non-volatile memory. A PLC is an example of a real time system since output results must be produced in response to input conditions within a bounded time, otherwise unintended operation will result.

History

The PLC was invented in response to the needs of the American automotive manufacturing industry. Programmable logic controllers were initially adopted by the automotive industry where software revision replaced the re-wiring of hard-wired control panels when production models changed.

Before the PLC, control, sequencing, and safety interlock logic for manufacturing automobiles was accomplished using hundreds or thousands of relays, cam timers, and drum sequencers and dedicated closed-loop controllers. The process for updating such facilities for the yearly model change-over was very time consuming and expensive, as electricians needed to individually rewire each and every relay.

In 1968 GM Hydramatic (the automatic transmission division of General Motors) issued a request for proposal for an electronic replacement for hard-wired relay systems. The winning proposal came from Bedford Associates of Bedford, Massachusetts. The first PLC, designated the 084 because it was Bedford Associates' eighty-fourth project, was the result. Bedford Associates started a new company dedicated to developing, manufacturing, selling, and servicing this new product: Modicon, which stood for MOdular DIgital CONtroller. One of the people who worked on that project was Dick Morley, who is considered to be the "father" of the PLC. The Modicon brand was sold in 1977 to Gould Electronics, and later acquired by German Company AEG and then by French Schneider Electric, the current owner.

One of the very first 084 models built is now on display at Modicon's headquarters in North Andover, Massachusetts. It was presented to Modicon by GM, when the unit was retired after nearly twenty years of uninterrupted service. Modicon used the 84 moniker at the end of its product range until the 984 made its appearance.

The automotive industry is still one of the largest users of PLCs.

Development

Early PLCs were designed to replace relay logic systems. These PLCs were programmed in "ladder logic", which strongly resembles a schematic diagram of relay logic. This program notation was chosen to reduce training demands for the existing technicians. Other early PLCs used a form of instruction list programming, based on a stack-based logic solver.

Modern PLCs can be programmed in a variety of ways, from ladder logic to more traditional programming languages such as BASIC and C. Another method is State Logic, a very high-level programming language designed to program PLCs based on state transition diagrams.

Many early PLCs did not have accompanying programming terminals that were capable of graphical representation of the logic, and so the logic was instead represented as a series of logic expressions in some version of Boolean format, similar to Boolean algebra. As programming terminals evolved, it became more common for ladder logic to be used, for the aforementioned reasons. Newer formats such as State Logic and Function Block (which is similar to the way logic is depicted when using digital integrated logic circuits) exist, but they are still not as popular as ladder logic. A primary reason for this is that PLCs solve the logic in a predictable and repeating sequence, and ladder logic allows the programmer (the person writing the logic) to see any issues with the timing of the logic sequence more easily than would be possible in other formats.

Programming

Early PLCs, up to the mid-1980s, were programmed using proprietary programming panels or special-purpose programming terminals, which often had dedicated function keys representing the various logical elements of PLC programs. Programs were stored on cassette tape cartridges. Facilities for printing and documentation were very minimal due to lack of memory capacity. The very oldest PLCs used non-volatile magnetic core memory.

More recently, PLCs are programmed using application software on personal computers. The computer is connected to the PLC through Ethernet, RS-232, RS-485 or RS-422 cabling. The programming software allows entry and editing of the ladder-style logic. Generally the software provides functions for debugging and troubleshooting the PLC software, for example, by highlighting portions of the logic to show current status during operation or via simulation. The software will upload and download the PLC program, for backup and restoration purposes. In some models of programmable controller, the program is transferred from a personal computer to the PLC though a programming board which writes the program into a removable chip such as an EEPROM or EPROM.

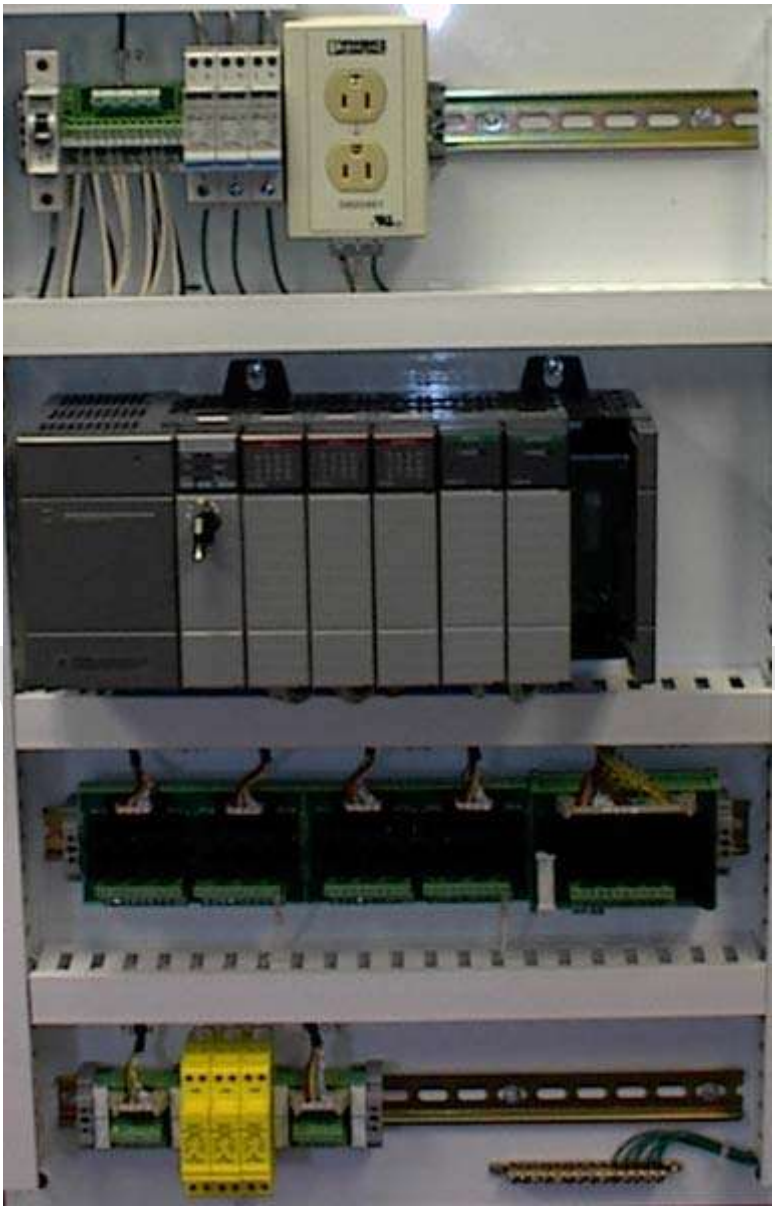
Functionality

The functionality of the PLC has evolved over the years to include sequential relay control, motion control, process control, distributed control systems and networking. The data handling, storage, processing power and communication capabilities of some modern PLCs are approximately equivalent to desktop computers. PLC-like programming combined with remote I/O hardware, allow a general-purpose desktop computer to overlap some PLCs in certain applications. Regarding the practicality of these desktop computer based logic controllers, it is important to note that they have not been generally accepted in heavy industry because the desktop computers run on less stable operating systems than do PLCs, and because the desktop computer hardware is typically not designed to the same levels of tolerance to temperature, humidity, vibration, and longevity as the processors used in PLCs. In addition to the hardware limitations of desktop based logic, operating systems such as Windows do not lend themselves to deterministic logic execution, with the result that the logic may not always respond to changes in logic state or input status with the extreme consistency in timing as is expected from PLCs. Still, such desktop logic applications find use in less critical situations, such as laboratory automation and use in small facilities where the application is less demanding and critical, because they are generally much less expensive than PLCs.

In more recent years, small products called PLRs (programmable logic relays), and also by similar names, have become more common and accepted. These are very much like PLCs, and are used in light industry where only a few points of I/O (i.e. a few signals coming in from the real world and a few going out) are involved, and low cost is desired. These small devices are typically made in a common physical size and shape by several manufacturers, and branded by the makers of larger PLCs to fill out their low end product range. Popular names include PICO Controller, NANO PLC, and other names implying very small controllers. Most of these have between 8 and 12 digital inputs, 4 and 8 digital outputs, and up to 2 analog inputs. Size is usually about 4" wide, 3" high, and 3" deep. Most such devices include a tiny postage stamp sized LCD screen for viewing simplified ladder logic (only a very small portion of the program being visible at a given time) and status of I/O points, and typically these screens are accompanied by a 4-way rocker push-button plus four more separate push-buttons, similar to the key buttons on a VCR remote control, and used to navigate and edit the logic. Most have a small plug for connecting via RS-232 or RS-485 to a personal computer so that programmers can use simple Windows applications for programming instead of being forced to use the tiny LCD and push-button set for this purpose. Unlike regular PLCs that are usually modular and greatly expandable, the PLRs are usually not modular or expandable, but their price can be two orders of magnitude less than a PLC and they still offer robust design and deterministic execution of the logic.

PLC Topics

Features



Control panel with PLC (grey elements in the center). The unit consists of separate elements, from left to right; power supply, controller, relay units for in- and output

The main difference from other computers is that PLCs are armored for severe conditions (such as dust, moisture, heat, cold) and have the facility for extensive input/output (I/O) arrangements. These connect the PLC to sensors and actuators. PLCs read limit switches, analog process variables (such as temperature and pressure), and the positions of complex positioning systems. Some use machine vision. On the actuator side, PLCs operate electric motors, pneumatic or hydraulic cylinders, magnetic relays, solenoids, or analog outputs. The input/output arrangements may be built into a simple PLC, or the PLC may have external I/O modules attached to a computer network that plugs into the PLC.

System scale

A small PLC will have a fixed number of connections built in for inputs and outputs. Typically, expansions are available if the base model has insufficient I/O.

Modular PLCs have a chassis (also called a rack) into which are placed modules with different functions. The processor and selection of I/O modules is customised for the particular application. Several racks can be administered by a single processor, and may have thousands of inputs and outputs. A special high speed serial I/O link is used so that racks can be distributed away from the processor, reducing the wiring costs for large plants.

User interface

PLCs may need to interact with people for the purpose of configuration, alarm reporting or everyday control.

A Human-Machine Interface (HMI) is employed for this purpose. HMIs are also referred to as MMIs (Man Machine Interface) and GUIs (Graphical User Interface).

A simple system may use buttons and lights to interact with the user. Text displays are available as well as graphical touch screens. More complex systems use a programming and monitoring software installed on a computer, with the PLC connected via a communication interface.

Communications

PLCs have built in communications ports, usually 9-pin RS-232, but optionally EIA-485 or Ethernet. Modbus, BACnet or DF1 is usually included as one of the communications protocols. Other options include various fieldbuses such as DeviceNet or Profibus. Other communications protocols that may be used are listed in the List of automation protocols.

Most modern PLCs can communicate over a network to some other system, such as a computer running a SCADA (Supervisory Control And Data Acquisition) system or web browser.

PLCs used in larger I/O systems may have peer-to-peer (P2P) communication between processors. This allows separate parts of a complex process to have individual control while allowing the subsystems to co-ordinate over the communication link. These communication links are also often used for HMI devices such as keypads or PC-type workstations.

Programming

PLC programs are typically written in a special application on a personal computer, then downloaded by a direct-connection cable or over a network to the PLC. The program is stored in the PLC either in battery-backed-up RAM or some other non-volatile flash memory. Often, a single PLC can be programmed to replace thousands of relays.

Under the IEC 61131-3 standard, PLCs can be programmed using standards-based programming languages. A graphical programming notation called Sequential Function Charts is available on certain programmable controllers. Initially most PLCs utilized Ladder Logic Diagram Programming, a model which emulated electromechanical control panel devices (such as the contact and coils of relays) which PLCs replaced. This model remains common today.

IEC 61131-3 currently defines five programming languages for programmable control systems: FBD (Function block diagram), LD (Ladder diagram), ST (Structured text, similar to the Pascal programming language), IL (Instruction list, similar to assembly language) and SFC (Sequential function chart). These techniques emphasize logical organization of operations.

While the fundamental concepts of PLC programming are common to all manufacturers, differences in I/O addressing, memory organization and instruction sets mean that PLC programs are never perfectly interchangeable between different makers. Even within the same product line of a single manufacturer, different models may not be directly compatible.

PLC compared with other control systems

Allen-Bradley PLC installed in a control panel

PLCs are well-adapted to a range of automation tasks. These are typically industrial processes in manufacturing where the cost of developing and maintaining the automation system is high relative to the total cost of the automation, and where changes to the system would be expected during its operational life. PLCs contain input and output devices compatible with industrial pilot devices and controls; little electrical design is required, and the design problem centers on expressing the desired sequence of operations. PLC applications are typically highly customized systems so the cost of a packaged PLC is low compared to the cost of a specific custom-built controller design. On the other hand, in the case of mass-produced goods, customized control systems are economic due to the lower cost of the components, which can be optimally chosen instead of a "generic" solution, and where the non-recurring engineering charges are spread over thousands or millions of units.

For high volume or very simple fixed automation tasks, different techniques are used. For example, a consumer dishwasher would be controlled by an electromechanical cam timer costing only a few dollars in production quantities.

A microcontroller-based design would be appropriate where hundreds or thousands of units will be produced and so the development cost (design of power supplies, input/output hardware and necessary testing and certification) can be spread over many sales, and where the end-user would not need to alter the control. Automotive applications are an example; millions of units are built each year, and very few end-users alter the programming of these controllers. However, some specialty vehicles such as transit busses economically use PLCs instead of custom-designed controls, because the volumes are low and the development cost would be uneconomic.

Very complex process control, such as used in the chemical industry, may require algorithms and performance beyond the capability of even high-performance PLCs. Very high-speed or precision controls may also require customized solutions; for example, aircraft flight controls.

Programmable controllers are widely used in motion control, positioning control and torque control. Some manufacturers produce motion control units to be integrated with PLC so that G-code (involving a CNC machine) can be used to instruct machine movements.

PLCs may include logic for single-variable feedback analog control loop, a "proportional, integral, derivative" or "PID controller". A PID loop could be used to control the temperature of a manufacturing process, for example. Historically PLCs were usually configured with only a few analog control loops; where processes required hundreds or thousands of loops, a distributed control system (DCS) would instead be used. As PLCs have become more powerful, the boundary between DCS and PLC applications has become less distinct.

PLCs have similar functionality as Remote Terminal Units. An RTU, however, usually does not support control algorithms or control loops. As hardware rapidly becomes more

powerful and cheaper, RTUs, PLCs and DCSs are increasingly beginning to overlap in responsibilities, and many vendors sell RTUs with PLC-like features and vice versa. The industry has standardized on the IEC 61131-3 functional block language for creating programs to run on RTUs and PLCs, although nearly all vendors also offer proprietary alternatives and associated development environments.

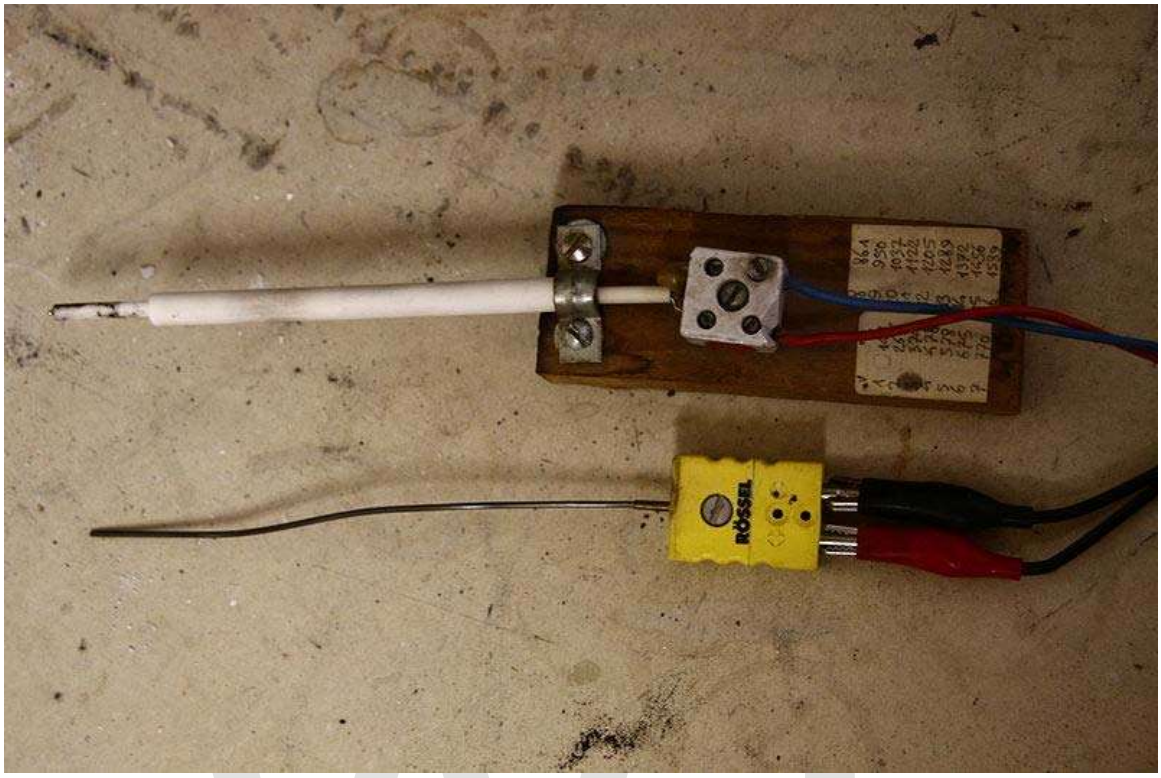
Digital and analog signals

Digital or discrete signals behave as binary switches, yielding simply an On or Off signal (1 or 0, True or False, respectively). Push buttons, limit switches, and photoelectric sensors are examples of devices providing a discrete signal. Discrete signals are sent using either voltage or current, where a specific range is designated as *On* and another as *Off*. For example, a PLC might use 24 V DC I/O, with values above 22 V DC representing *On*, values below 2VDC representing *Off*, and intermediate values undefined. Initially, PLCs had only discrete I/O.

Analog signals are like volume controls, with a range of values between zero and full-scale. These are typically interpreted as integer values (counts) by the PLC, with various ranges of accuracy depending on the device and the number of bits available to store the data. As PLCs typically use 16-bit signed binary processors, the integer values are limited between -32,768 and +32,767. Pressure, temperature, flow, and weight are often represented by analog signals. Analog signals can use voltage or current with a magnitude proportional to the value of the process signal. For example, an analog 0 - 10 V input or 4-20 mA would be converted into an integer value of 0 - 32767.

Current inputs are less sensitive to electrical noise (i.e. from welders or electric motor starts) than voltage inputs

Sensor



Thermocouple sensor for high temperature measurement

A **sensor** is a device that measures a physical quantity and converts it into a signal which can be read by an observer or by an instrument. For example, a mercury-in-glass thermometer converts the measured temperature into expansion and contraction of a liquid which can be read on a calibrated glass tube. A thermocouple converts temperature to an output voltage which can be read by a voltmeter. For accuracy, most sensors are calibrated against known standards.

Use

Sensors are used in everyday objects such as touch-sensitive elevator buttons (tactile sensor) and lamps which dim or brighten by touching the base. There are also innumerable applications for sensors of which most people are never aware. Applications include cars, machines, aerospace, medicine, manufacturing and robotics.

A sensor is a device which receives and responds to a signal. A sensor's sensitivity indicates how much the sensor's output changes when the measured quantity changes. For instance, if the mercury in a thermometer moves 1 cm when the temperature changes by 1 °C, the sensitivity is 1 cm/°C (it is basically the slope Dy/Dx assuming a linear characteristic). Sensors that measure very small changes must have very high sensitivities. Sensors also have an impact on what they measure; for instance, a room temperature thermometer inserted into a hot cup of liquid cools the liquid while the liquid heats the thermometer. Sensors need to be designed to have a small effect on what is

measured, making the sensor smaller often improves this and may introduce other advantages. Technological progress allows more and more sensors to be manufactured on a microscopic scale as microsensors using MEMS technology. In most cases, a microsensor reaches a significantly higher speed and sensitivity compared with macroscopic approaches.

Classification of measurement errors

A good sensor obeys the following rules:

- Is sensitive to the measured property
- Is insensitive to any other property likely to be encountered in its application
- Does not influence the measured property

Ideal sensors are designed to be linear or linear to some simple mathematical function of the measurement, typically logarithmic. The output signal of such a sensor is linearly proportional to the value or simple function of the measured property. The sensitivity is then defined as the ratio between output signal and measured property. For example, if a sensor measures temperature and has a voltage output, the sensitivity is a constant with the unit [V/K]; this sensor is linear because the ratio is constant at all points of measurement.

Sensor deviations

If the sensor is not ideal, several types of deviations can be observed:

- The sensitivity may in practice differ from the value specified. This is called a sensitivity error, but the sensor is still linear.
- Since the range of the output signal is always limited, the output signal will eventually reach a minimum or maximum when the measured property exceeds the limits. The full scale range defines the maximum and minimum values of the measured property.
- If the output signal is not zero when the measured property is zero, the sensor has an offset or bias. This is defined as the output of the sensor at zero input.
- If the sensitivity is not constant over the range of the sensor, this is called nonlinearity. Usually this is defined by the amount the output differs from ideal behavior over the full range of the sensor, often noted as a percentage of the full range.
- If the deviation is caused by a rapid change of the measured property over time, there is a dynamic error. Often, this behaviour is described with a bode plot showing sensitivity error and phase shift as function of the frequency of a periodic input signal.
- If the output signal slowly changes independent of the measured property, this is defined as drift (telecommunication).
- Long term drift usually indicates a slow degradation of sensor properties over a long period of time.

- Noise is a random deviation of the signal that varies in time.
- Hysteresis is an error caused by when the measured property reverses direction, but there is some finite lag in time for the sensor to respond, creating a different offset error in one direction than in the other.
- If the sensor has a digital output, the output is essentially an approximation of the measured property. The approximation error is also called digitization error.
- If the signal is monitored digitally, limitation of the sampling frequency also can cause a dynamic error, or if the variable or added noise noise changes periodically at a frequency near a multiple of the sampling rate may induce aliasing errors.
- The sensor may to some extent be sensitive to properties other than the property being measured. For example, most sensors are influenced by the temperature of their environment.

All these deviations can be classified as systematic errors or random errors. Systematic errors can sometimes be compensated for by means of some kind of calibration strategy. Noise is a random error that can be reduced by signal processing, such as filtering, usually at the expense of the dynamic behaviour of the sensor.

Resolution

The resolution of a sensor is the smallest change it can detect in the quantity that it is measuring. Often in a digital display, the least significant digit will fluctuate, indicating that changes of that magnitude are only just resolved. The resolution is related to the precision with which the measurement is made. For example, a scanning tunneling probe (a fine tip near a surface collects an electron tunnelling current) can resolve atoms and molecules.actuator is something that converts energy into motion

Types

Sensors in Nature

All living organisms contain biological sensors with functions similar to those of the mechanical devices described. Most of these are specialized cells that are sensitive to:

- Light, motion, temperature, magnetic fields, gravity, humidity, vibration, pressure, electrical fields, sound, and other physical aspects of the external environment
- Physical aspects of the internal environment, such as stretch, motion of the organism, and position of appendages (proprioception)
- Environmental molecules, including toxins, nutrients, and pheromones
- Estimation of biomolecules interaction and some kinetics parameters
- Internal metabolic milieu, such as glucose level, oxygen level, or osmolality
- Internal signal molecules, such as hormones, neurotransmitters, and cytokines
- Differences between proteins of the organism itself and of the environment or alien creatures.

Biosensor

In biomedicine and biotechnology, sensors which detect analytes thanks to a biological component, such as cells, protein, nucleic acid or biomimetic polymers, are called biosensors. Whereas a non-biological sensor, even organic (=carbon chemistry), for biological analytes is referred to as sensor or nanosensor (such a microcantilevers). This terminology applies for both in vitro and in vivo applications. The encapsulation of the biological component in biosensors, presents with a slightly different problem that ordinary sensors, this can either be done by means of a semipermeable barrier, such as a dialysis membrane or a hydrogel, a 3D polymer matrix, which either physically constrains the sensing macromolecule or chemically (macromolecule is bound to the scaffold).

Actuator

An **actuator** is a mechanical device for moving or controlling a mechanism or system. It is operated by a source of energy, usually in the form of an electric current, hydraulic fluid pressure or pneumatic pressure, and converts that energy into some kind of motion.

Examples and applications

- Mechanical actuators operate by conversion of rotary motion into linear motion, or vice versa. Conversion is commonly made via a few simple types of mechanism including:
    - **Screw**: Screw jack, ball screw and roller screw actuators all operate on the principle of the simple machine known as the screw. By rotating the actuator's nut, the screw shaft moves in a line. By moving the screw shaft, the nut rotates.
    - **Wheel and axle**: Hoist, winch, rack and pinion, chain drive, belt drive, rigid chain and rigid belt actuators operate on the principle of the wheel and axle. By rotating a wheel/axle (e.g. drum, gear, pulley or shaft) a linear member (e.g. cable, rack, chain or belt) moves. By moving the linear member, the wheel/axle rotates.
- In engineering, actuators are frequently used as mechanisms to introduce motion, or to clamp an object so as to prevent motion. In electronic engineering, actuators are a subdivision of transducers. They are devices which transform an input signal (mainly an electrical signal) into motion. Specific examples include: electrical motors, pneumatic actuators, hydraulic actuators, linear actuators, comb drive, piezoelectric actuators and amplified piezoelectric actuators, thermal bimorphs, micromirror devices and electroactive polymers.
- Motors are mostly used when circular motions are needed, but can also be used for linear applications by transforming circular to linear motion with a bolt and

screw transducer. On the other hand, some actuators are intrinsically linear, such as piezoelectric actuators.

A centralized controller can be used, or multiple intelligent devices can be distributed around the home.

Interconnection

By wire:

1. optical fiber
2. cable (coaxial and twisted pair) , including:

   xDSL

3. powerline, including:

   X10
   Universal powerline bus (UPB)

Wireless:

1. radio frequency, including:

   Wi-Fi
   GPRS and UMTS
   Bluetooth
   DECT
   ZigBee
   Z-Wave
   ONE-NET
   EnOcean

2. infrared, including:

   Consumer IR

Both Wireless and Wire

1. INSTEON

Classifications of domestic network technologies

- Device interconnection:
  - Bluetooth
  - IEEE 1394 interface (FireWire)

    - IrDA
    - Universal Serial Bus (USB)
- Control and automation nets:
    - SCS BUS with OpenWebNet
    - C-Bus (protocol)
    - CEBus
    - EnOcean
    - EHS
    - INSTEON
    - KNX (European Installation Bus)
    - LonWorks
    - ONE-NET
    - Universal Powerline Bus
    - X10
    - ZigBee
- Data nets:
    - Ethernet
    - Homeplug
    - HomePNA
    - WiFi

There have been many attempts to standardise the forms of hardware, electronic and communication interfaces needed to construct a home automation system. Some standards use additional communication and control wiring, some embed signals in the existing power circuit of the house, some use radio frequency (RF) signals, and some use a combination of several methods. Control wiring is hardest to retrofit into an existing house. Some appliances include USB that is used to control it and connect it to a domotics network. Bridges translate information from one standard to another, *e.g.*, from X10 to European Installation Bus.

Tasks

HVAC

Heating, Ventilation and Air Conditioning (HVAC) solutions include temperature and humidity control. This is generally one of the most important aspects to a homeowner. An Internet-controlled thermostat, for example, can both save money and help the environment, by allowing the homeowner to control the building's heating and air conditioning systems remotely.

Lighting

Lighting control systems can be used to control household electric lights.

- Extinguish all the lights of the house

- Replace manual switching with Automation of on and off signals for any or all lights
- Regulation of electric illumination levels according to the level of ambient light available
- Change the ambient colour of lighting via control of LEDs or electronic dimmers

Natural lighting control involves controlling window shades, LCD shades, draperies and awnings.

Audio and video

This category includes audio and video switching and distribution. Multiple audio or video sources can be selected and distributed to one or more rooms.

Security

Control and integration of security systems.

With Home Automation, the consumer can select and watch cameras live from an Internet source to their home or business. Security cameras can be controlled, allowing the user to observe activity around a house or business right from a Monitor or touch panel. Security systems can include motion sensors that will detect any kind of unauthorized movement and notify the user through the security system or via cell phone.

This category also includes control and distribution of security cameras.

- Detection of possible intrusion
  - sensors of detection of movement
  - sensors of magnetic contact of door/window
  - sensors of glass breaking
  - sensors of pressure changes
- Simulation of presence.
- Detection of fire, gas leaks, water leaks
- Medical alert. Teleassistance.
- Precise and safe closing of blinds.

Intercoms

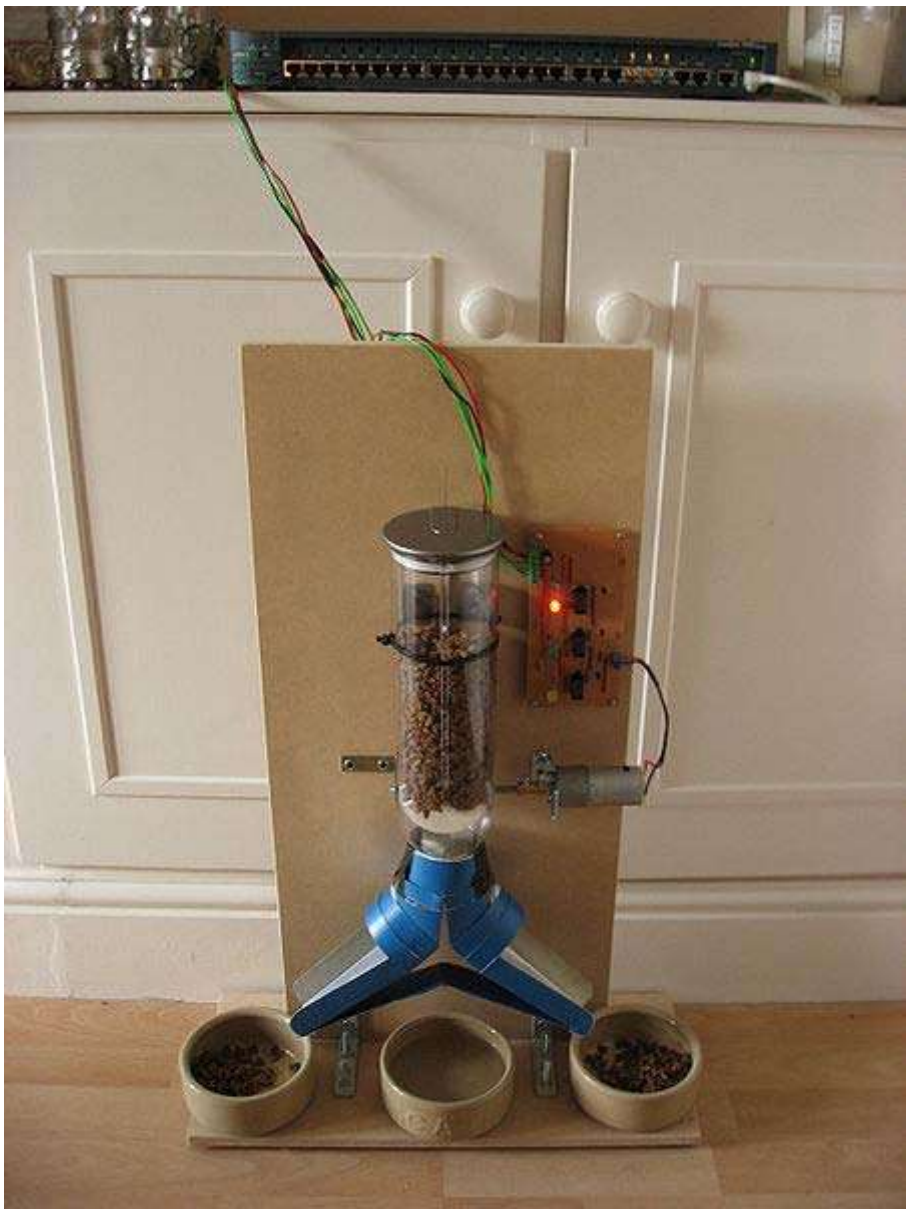
An intercom system allows communication via a microphone and loud speaker between multiple rooms. Integration of the intercom to the telephone, or of the video door entry system to the television set, allowing the residents to view the door camera automatically.

Robotics

- Control of home robots, using if necessary domotic electric beacon.

- Home robot communication (i.e. using WiFi) with the domotic network and other home robots.

Other systems



A homemade Internet-enabled cat feeder.

Using special hardware, almost any device can be monitored and controlled automatically or remotely, including:

- Coffee pot
- Garage door
- Pet feeding and watering

- Plant watering
- Pool pump(s) and heater, Hot tub and Spa
- Sump Pump

Costs

An automated home can be a very simple grouping of controls, or it can be heavily automated where any appliance that is plugged into electrical power is remotely controlled. Costs mainly include equipment, components, furniture, and custom installation.

Ongoing costs include electricity to run the control systems, maintenance costs for the control and networking systems, including troubleshooting, and eventual cost of upgrading as standards change. Increased complexity may also increase maintenance costs for networked devices.

Learning to use a complex system effectively may take significant time and training.

Control system security may be difficult and costly to maintain, especially if the control system extends beyond the home, for instance by wireless or by connection to the internet or other networks.

Smart Grid

Home automation technologies are viewed as integral additions to the Smart grid. The ability to control lighting, appliances, HVAC as well as Smart Grid applications (load shedding, demand response, real-time power usage and price reporting) will become vital as Smart Grid initiatives are rolled out. Green Automation is the term coined to describe energy management strategies in home automation when data from smart grids is combined with home automation systems to use resources either at there cheapest prices or most available. For example taking advantage of high solar panel out put in the middle of the day to run washing machines automatically.

Chapter- 6

Interactive Voice Response

Interactive Voice Response (**IVR**) is a technology that allows a computer to interact with humans through the use of voice and DTMF keypad inputs.

In telecommunications, IVR allows customers to interact with a company's database via a telephone keypad or by speech recognition, after which they can service their own inquiries by following the IVR dialogue. IVR systems can respond with prerecorded or dynamically generated audio to further direct users on how to proceed. IVR applications can be used to control almost any function where the interface can be broken down into a series of simple interactions. IVR systems deployed in the network are sized to handle large call volumes.

IVR technology is also being introduced into automobile systems for hands-free operation. Current deployment in automobiles revolves around satellite navigation, audio and mobile phone systems.

It has become common in industries that have recently entered the telecommunications industry to refer to an Automated Attendant as an IVR. The terms **Automated Attendant** and **IVR** are distinct and mean different things to traditional telecommunications professionals, whereas emerging telephony and VoIP professionals often use the term **IVR** as a catch-all to signify any kind of telephony menu, even a basic automated attendant. The term **VRU**, for **Voice Response Unit**, is sometimes used as well.

History

The blueprint for IVR began in 1961, when the Bell System developed a new tone dialing methodology. Bell unveiled the first telephone that could dial area codes using DTMF technology at the Seattle World Fair in 1962. DTMF telephones enabled the use of in-band signaling, i.e., they transmit audible tones in the same 300 Hz to 3.4 KHz range occupied by the human voice.

Despite the fact that more companies began using IVR technology in the 1970's to automate tasks in call centers, the technology was still costly and complicated. Early

speech recognition systems were DSP technology based, and were limited to small vocabularies. However, by the 1980s a number of new competitors entered the market and uptake of IVR technology started to increase. As speech recognition software developed the technology changed from DSP to a client server architecture.

As call centers began to migrate to multimedia in the late 1990s, companies started to invest in Computer Telephony Integration (CTI) with IVR systems. IVR became vital for call centers deploying universal queuing and routing solutions and acted as an agent which collected customer data to enable intelligent routing decisions.

In the subsequent decade, speech recognition started to become more common and cheaper to deploy. This was due to increased CPU power and the migration of speech applications from proprietary code to the VXML standard.

Typical uses

IVR systems are typically used to service high call volumes, reduce cost and improve the customer experience. Examples of typical IVR applications are telephone banking, televoting, and credit card services. Companies also use IVR services to extend their business hours to 24/7 operation.

The use of IVR and voice automation enables a company to improve its customer service and lower its costs, due to the fact that callers' queries can be resolved without the need for queueing and incurring the cost of a live agent who, in turn, can be directed to deal with more demanding areas of the service. If the caller does not find the information they need, or requires further assistance, the call can then be transferred to an agent. This makes for a more efficient system in which agents have more time to deal with complex interactions: for example, customer retention, up selling, cross selling and issue resolution. This way, the customer is more likely to be satisfied with a personalized service and the interaction is likely to be more fulfilling and rewarding for the agent, as opposed to dealing with basic inquiries that require yes/no responses, such as obtaining customer details. Employee satisfaction is important in the telecommunications industry due to the fast turnover of staff - IVR is therefore one way of retaining the workforce and allowing them to do a more effective job.

Call centers use IVR systems to identify and segment callers. The ability to identify customers allows services to be tailored according to the customer profile. The caller can be given the option to wait in the queue, choose an automated service, or request a callback (at a suitable time and telephone number). The system may obtain caller line identification (CLI) data from the network to identify the caller. This is currently available for about 80% of inbound calls. In the cases where CLI is withheld or unavailable, the caller can be asked to identify themselves by other methods such as a PIN or password. The use of DNIS ensures that the correct application and language is executed by the IVR system.

IVR also enables customer prioritization. In a system wherein individual customers may have a different status the service will automatically prioritize the individual's call and move prime customers to the front of the calling queue.

CTI allows a contact center or organization to gather information about the caller as a means of directing their inquiry to the appropriate agent. CTI can transfer relevant information about the individual customer from the database to the agent desktop using a screen-pop, making for a more effective and efficient service.

IVR may be used by survey organizations for asking more sensitive questions where the investigators are concerned that a respondent might feel less comfortable providing these answers to a human interlocutor (such as questions about drug use or sexual behavior). In some cases an IVR system can be used in the same survey in conjunction with a human interviewer. For example, during the survey the interviewer might inform the respondent that for the next series of questions they will be sent to an IVR system to continue or complete the interview.

Voice-Activated Dialling

Voice-Activated Dialling (VAD) IVR systems are used to automate routine enquiries to switchboard or PABX (Private Automatic Branch eXchange) operators, and are used in many hospitals and large businesses to reduce the caller waiting time. An additional function is the ability to allow external callers to page staff and transfer the inbound call to the paged person.

Entertainment and information

Some of the largest installed IVR platforms are used for televoting on television game shows, such as *Pop Idol* and *Big Brother*, which can generate enormous call spikes. Often, the network provider will have to deploy *call gapping* in the PSTN to prevent network overload.

Anonymous access

IVR systems allow callers to obtain data relatively anonymously. Hospitals and clinics have used IVR systems to allow callers to receive anonymous access to test results. This is information that could easily be handled by a person but the IVR system is used to preserve privacy and avoid potential embarrassment of sensitive information or test results. Users are given a passcode to access their results.

Clinical trials

IVR systems are used by pharmaceutical companies and contract research organizations to conduct clinical trials and manage the large volumes of data generated. The caller will respond to questions in their preferred language and their responses will be logged into a database and possibly recorded at the same time to confirm authenticity. Applications

include patient randomization and drug supply management. They are also used in recording patient diaries and questionnaires.

Outbound calling

IVR systems can be used for outbound calls, as IVR systems are more intelligent than many predictive dialer systems, and can use Call Progress Detection to recognize different line conditions as follows:

- Answer (the IVR can tell the customer who is calling and ask them to wait for an agent)
- Answered by voice mail or answering machine (in these circumstances the IVR system can leave a message)
- Fax tone (the IVR can leave a TIFF image fax message)
- Divert messages (the IVR will abandon the call)
- No answer

Other uses

Other common IVR services include:

- Mobile — Pay-As-You-Go account funding; registration; mobile purchases, such as ring tones and logos
- Banking — balance, payments, transfers, transaction history
- Retail & Entertainment — orders, bookings, credit & debit card payments
- Utilities — meter readings
- Travel — ticket booking, flight information, check-in
- Adult entertainment — dating, chat line
- Weather forecasts

Technologies used

DTMF signals (entered via the telephone keypad) and natural language speech recognition interpret the caller's response to voice prompts.

Other technologies include the ability to speak complex and dynamic information, such as an e-mail, news report or weather information using Text-To-Speech (TTS). TTS is computer generated synthesized speech that is no longer the robotic voice generally associated with computers. Real voices create the speech in fragments that are spliced together (concatenated) and smoothed before being played to the caller.

An IVR can be deployed in several different ways:

1. Equipment installed on the customer premises
2. Equipment installed in the PSTN (Public Switched Telephone Network)
3. Application service provider (ASP) / Hosted IVR

IVR can be used to provide a more sophisticated voice mail experience to the caller. For example, the IVR could ask if the caller wishes to hear, edit, forward or remove a message.

An automatic call distributor (ACD) is often the first point of contact when calling many larger businesses. An ACD uses digital storage devices to play greetings or announcements, but typically routes a caller without prompting for input. An IVR can play announcements and request an input from the caller. This information can be used to profile the caller and route the call to an agent with a particular skill set. (A skill set is a function applied to a group of call-center agents with a particular skill.)

Interactive voice response can be used to front-end a call center operation by identifying the needs of the caller. Information can be obtained from the caller such as an account number. Answers to simple questions such as account balances or pre-recorded information can be provided without operator intervention. Account numbers from the IVR are often compared to caller ID data for security reasons and additional IVR responses are required if the caller ID does not match the account record.

IVR call flows are created in a variety of ways. A traditional IVR depended upon proprietary programming or scripting languages, whereas modern IVR applications are generated in a similar way to Web pages, using standards such as VoiceXML, CCXML, SRGS and SSML. The ability to use XML-driven applications allows a Web server to act as the application server, freeing the IVR developer to focus on the call flow. It was widely believed that developers would no longer require specialized programming skills; however, this has been proven to be misguided as IVR applications need to understand the human reaction to the application dialog.

Higher level IVR development tools are available to further simplify the application development process. A call flow diagram can be drawn with a GUI tool and the presentation layer (typically VoiceXML) can be automatically generated. In addition, these tools normally provide extension mechanisms for software integration, such as an HTTP interface to a Web site and a Java interface for connecting to a database.

In telecommunications, an **audio response unit** (ARU) is a device that provides synthesized voice responses to DTMF keypresses by processing calls based on (a) the call-originator input, (b) information received from a database, and (c) information in the incoming call, such as the time of day.

ARUs increase the number of information calls handled and to provide consistent quality in information retrieval.

Outsourcing vs. contact center automation

Contact centers can be expensive to run, and are often seen as cost centers; however, the ability to up-sell services and products to customers can offset operational expenditure, and effectively reduce the average cost per call handled.

Methods of reducing contact center running costs include outsourcing and automation. Outsourcing to other countries can reduce operational expenditure by as much as 30%, however, differences in culture and language can prove problematic for customers, whose dissatisfaction can lead to customer complaints and loss of business. Also, it is more difficult to up-sell to customers from foreign contact centers.

Automation in a contact center can also reduce operational expenditure by around 30% though the introduction of technologies such as customer profiling, CTI, and IVR using speech recognition. The use of automation in the contact center promotes efficiency, allowing contact centers to be located in the country from which the call is originated. Customer satisfaction can be monitored by the use of customer survey applications. The information from survey applications can be used to improve customer service.

VoIP

The increased usage of VoIP in voice networks is likely to affect how IVR will be used in voice networks, this is due to the introduction of protocols such as Session Initiation Protocol (SIP). The introduction of SIP means that point to point communications is no longer restricted to voice calls but can now be extended to multimedia technologies such as video. This will bring a new meaning to automated services as IVR extends its reach to video calls. Many IVR manufacturers are currently working on IVVR (Interactive Voice and Video Response) systems, especially for the mobile phone networks. The use of video will give IVR systems the ability to use graphical and video information to assist the caller.

The introduction of video IVR may allow systems in the future the ability to read emotions and facial expressions. It may be used to identify the caller, using technology such as Iris scan or other biometric means. Recordings of the caller may be stored to monitor certain transactions, and may be used to reduce identity fraud.

Unified communications in the SIP contact center

With the introduction of SIP contact centers, automation has finally come of age. Call control in a SIP contact center can be implemented by CCXML scripting, which is an adjunct to the VXML language used to generate modern IVR dialogues. As calls are queued in the SIP contact center, the IVR system can provide treatment or automation, wait for a fixed period, or play music. Inbound calls to a SIP contact center must be queued or terminated against a SIP end point; SIP IVR systems can be used to replace agents directly by the use of Applications deployed using BBUA (Back to Back User Agents).

Interactive Messaging Response (IMR)

As communications have migrated to multimedia so has Automation. The introduction of Instant Messaging (IM) in Contact Centers is starting to take off. Agents can handle up to 6 different IM conversations at the same time and so agent productivity is increasing.

IVR Technology is being used to automate IM Conversations using existing Speech Recognition Software. This is different from email handling as email automated response is based on key word spotting. IM conversations are different to email as IM is conversational. The use of text messaging abbreviations and smilies requires different grammars than those currently used for speech recognition. IM is also starting to replace text messaging on Multimedia Mobile handsets and is expected to become more widely used.

Hosted vs. on-premise IVR

With the introduction of Web services into the Contact Center, Host integration has been simplified. The use of Web based applications allow IVR applications to be hosted remotely from the contact center. This allows the use of hosted IVR applications using speech to be made available to smaller Contact Centers across the globe and is likely to lead to an expansion of ASP (Application Service Providers).

IVR applications can also be hosted in the public network, which do not require contact center integration. This will include public announcement messages or message services for small business. It is also possible to use two prong IVR services where the initial IVR application is used to route the call to the appropriate contact center. This can be used to balance loading across multiple contact centers or provide business continuity in the event of system outage.

Criticism

IVR is sometimes criticized as being unhelpful and difficult to use due to poor design and lack of appreciation of the caller's needs. Some callers object to providing voice response to an automated system and prefer speaking with a human respondent.

Companies have also been criticized for using IVR to reduce operational costs but not offering similar services using agents. Such methods tend to frustrate customers who feel that their right to speak to an agent is being restricted. Examples of services criticized in this way include debt recovery and giveaways (concert tickets, satellite/cable receivers, etc.).