# UNIT 5

## Natural Language Discourse Structure

Using world knowledge - Discourse structure - Defining conversational agent - An introduction to logic model - Theoretic semantics - Symbolic computation - Speech recognition and spoken Language, Applications: Machine Translation, Information Retrieval.

### Using world knowledge - Discourse structure

The most difficult problem of AI is to process the natural language by computers or in other words *natural language processing* is the most difficult problem of artificial intelligence. If we talk about the major problems in NLP, then one of the major problems in NLP is discourse processing − building theories and models of how utterances stick together to form coherent discourse. Actually, the language always consists of collocated, structured and coherent groups of sentences rather than isolated and unrelated sentences like movies. These coherent groups of sentences are referred to as discourse.

### Concept of Coherence

Coherence and discourse structure are interconnected in many ways. Coherence, along with property of good text, is used to evaluate the output quality of natural language generation system. The question that arises here is what does it mean for a text to be coherent? Suppose we collected one sentence from every page of the newspaper, then will it be a discourse? Of-course, not. It is because these sentences do not exhibit coherence. The coherent discourse must possess the following properties −

### Coherence relation between utterances

The discourse would be coherent if it has meaningful connections between its utterances. This property is called coherence relation. For example, some sort of explanation must be there to justify the connection between utterances.

### Relationship between entities

Another property that makes a discourse coherent is that there must be a certain kind of relationship with the entities. Such kind of coherence is called entity-based coherence.

Discourse structure

An important question regarding discourse is what kind of structure the discourse must have. The answer to this question depends upon the segmentation we applied on discourse. Discourse segmentations may be defined as determining the types of structures for large discourse. It is quite difficult to implement discourse segmentation, but it is very important for **information retrieval, text summarization and information extraction** kind of applications.

Algorithms for Discourse Segmentation

In this section, we will learn about the algorithms for discourse segmentation. The algorithms are described below −

## Unsupervised Discourse Segmentation

The class of unsupervised discourse segmentation is often represented as linear segmentation. We can understand the task of linear segmentation with the help of an example. In the example, there is a task of segmenting the text into multi-paragraph units; the units represent the passage of the original text. These algorithms are dependent on cohesion that may be defined as the use of certain linguistic devices to tie the textual units together. On the other hand, lexicon cohesion is the cohesion that is indicated by the relationship between two or more words in two units like the use of synonyms.

## Supervised Discourse Segmentation

The earlier method does not have any hand-labeled segment boundaries. On the other hand, supervised discourse segmentation needs to have boundary-labeled training data. It is very easy to acquire the same. In supervised discourse segmentation, discourse marker or cue words play an important role. Discourse marker or cue word is a word or phrase that functions to signal discourse structure. These discourse markers are domain-specific.

## Text Coherence

Lexical repetition is a way to find the structure in a discourse, but it does not satisfy the requirement of being coherent discourse. To achieve the coherent discourse, we must focus on coherence relations in specific. As we know that coherence relation defines the possible connection between utterances in a discourse. Hebb has proposed such kind of relations as follows −

We are taking two terms $\mathbf{S_0}$ and $\mathbf{S_1}$ to represent the meaning of the two related sentences –

### Result

It infers that the state asserted by $S_1$ could cause the state asserted by $S_0$. For example, two statements show the relationship − Ram fought with Sham's friend. He was drunk.

### Explanation

It infers that the state asserted by $\mathbf{S_1}$ could cause the state asserted by $\mathbf{S_0}$. For example, two statements show the relationship − Ram fought with Shyam's friend. He was drunk.

### Parallel

It infers p(a1,a2,…) from assertion of $S_0$ and p(b1,b2,…) from assertion $S_1$. Here ai and bi are similar for all i. For example, two statements are parallel − Ram wanted car. Shyam wanted money.

### Elaboration

It infers the same proposition P from both the assertions − $S_0$ and $S_1$ For example, two statements show the relation elaboration: Ram was from Chandigarh. Shyam was from Kerala.

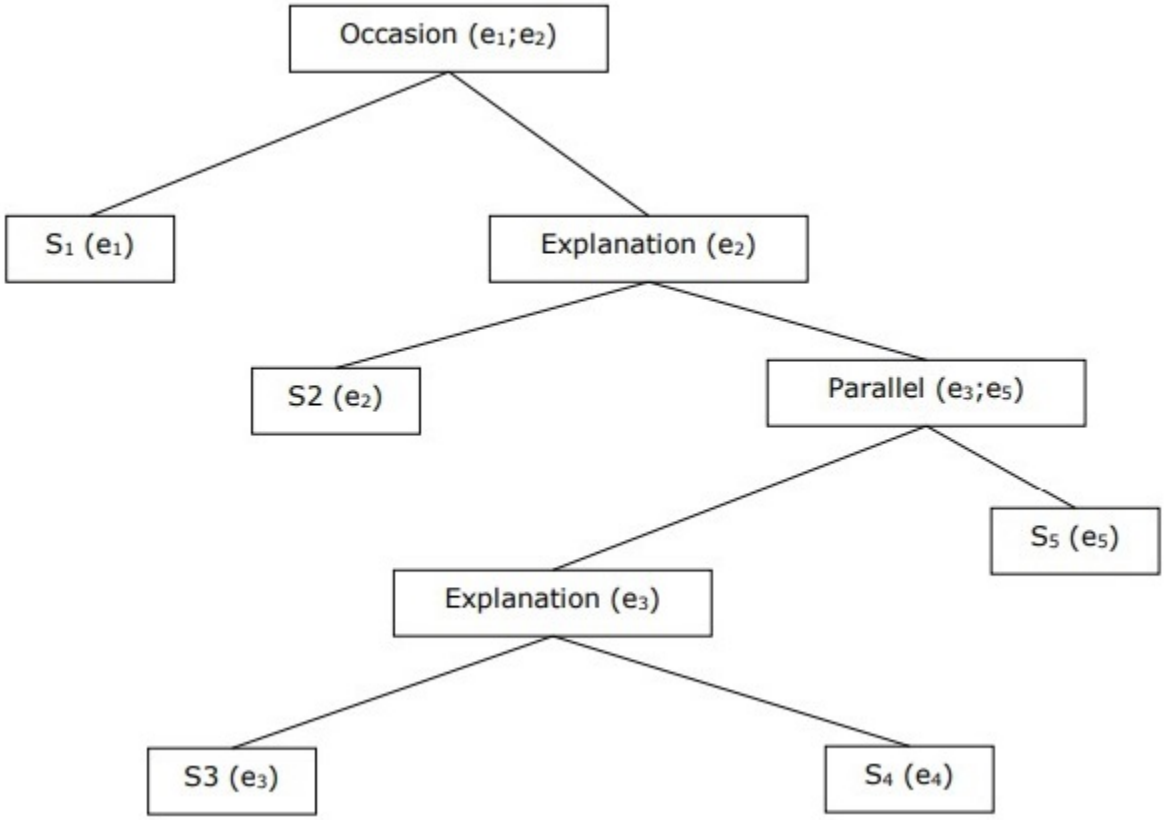
### Occasion

It happens when a change of state can be inferred from the assertion of $\mathbf{S_0}$, final state of which can be inferred from $\mathbf{S_1}$ and vice-versa. For example, the two statements show the relation occasion: Ram picked up the book. He gave it to Shyam.

**Building Hierarchical Discourse Structure**

The coherence of entire discourse can also be considered by hierarchical structure between coherence relations. For example, the following passage can be represented as hierarchical structure −

- **S1** − Ram went to the bank to deposit money.
- **S2** − He then took a train to Shyam's cloth shop.
- **S3** − He wanted to buy some clothes.
- **S4** − He do not have new clothes for party.
- **S5** − He also wanted to talk to Shyam regarding his health




**Reference Resolution**

Interpretation of the sentences from any discourse is another important task and to achieve this we need to know who or what entity is being talked about. Here, interpretation reference is the key element. **Reference** may be defined as the linguistic expression to denote an entity or individual. For example, in the passage, Ram, the manager of ABC bank, saw his friend Shyam at a shop. He went to meet him, the linguistic expressions like Ram, His, He are reference.

On the same note, **reference resolution** may be defined as the task of determining what entities are referred to by which linguistic expression.

**Terminology Used in Reference Resolution**

We use the following terminologies in reference resolution −

- **Referring expression** − The natural language expression that is used to perform reference is called a referring expression. For example, the passage used above is a referring expression.
- **Referent** − It is the entity that is referred. For example, in the last given example Ram is a referent.
- **Corefer** − When two expressions are used to refer to the same entity, they are called corefers. For example, ***Ram*** and ***he*** are corefers.
- **Antecedent** − The term has the license to use another term. For example, ***Ram*** is the antecedent of the reference ***he***.
- **Anaphora & Anaphoric** − It may be defined as the reference to an entity that has been previously introduced into the sentence. And, the referring expression is called anaphoric.
- **Discourse model** − The model that contains the representations of the entities that have been referred to in the discourse and the relationship they are engaged in.

Types of Referring Expressions

Let us now see the different types of referring expressions. The five types of referring expressions are described below −

Indefinite Noun Phrases

Such kind of reference represents the entities that are new to the hearer into the discourse context. For example − in the sentence Ram had gone around one day to bring him some food − some is an indefinite reference.

Definite Noun Phrases

Opposite to above, such kind of reference represents the entities that are not new or identifiable to the hearer into the discourse context. For example, in the sentence - I used to read The Times of India – The Times of India is a definite reference.

Pronouns

It is a form of definite reference. For example, Ram laughed as loud as he could. The word **he** represents pronoun referring expression.

Demonstratives

These demonstrate and behave differently than simple definite pronouns. For example, this and that are demonstrative pronouns.

Names

It is the simplest type of referring expression. It can be the name of a person, organization and location also. For example, in the above examples, Ram is the name-refereeing expression.

Reference Resolution Tasks

The two reference resolution tasks are described below.

Coreference Resolution

It is the task of finding referring expressions in a text that refer to the same entity. In simple words, it is the task of finding corefer expressions. A set of coreferring expressions are called coreference chain. For example - He, Chief Manager and His - these are referring expressions in the first passage given as example.

Constraint on Coreference Resolution

In English, the main problem for coreference resolution is the pronoun it. The reason behind this is that the pronoun it has many uses. For example, it can refer much like he and she. The pronoun it also refers to the things that do not refer to specific things. For example, It's raining. It is really good.

Pronominal Anaphora Resolution

Unlike the coreference resolution, pronominal anaphora resolution may be defined as the task of finding the antecedent for a single pronoun. For example, the pronoun is his and the task of pronominal anaphora resolution is to find the word Ram because Ram is the antecedent.

**What is a Conversational Agent?**

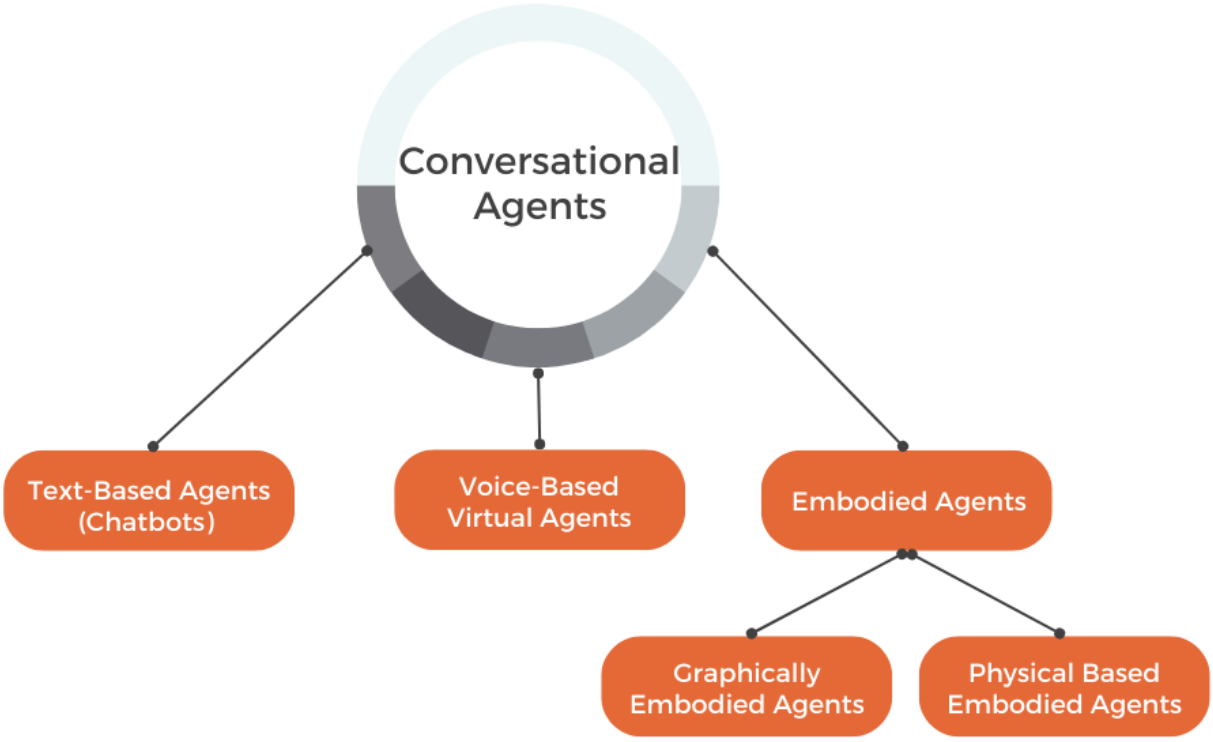
A conversational agent is a program designed to converse with humans using natural language.
A conversational agent is any dialogue system that conducts natural language processing (NLP) and responds automatically using human language. Conversational agents represent the practical implementation of computational linguistics, and are usually deployed as Chabot and virtual or AI assistants.

A conversational agent is a virtual agent you can use to communicate with a human in natural language. It simulates human-to-human interaction and understands context and meaning just as humans do.

It can talk to people on phones, computers, and other devices, allowing them to order food or do other functions through voice, text, or chat. It can achieve these using technologies like natural language processing (NLP), machine learning (ML), speech recognition, text-to-speech synthesis, and dialog management to interact with people through various mediums.

In this article, we'll discuss how they differ from Chabot, how they work, and how they can be applied in different industries.

**How does a conversational agent work?**




**Use cases of a conversational agent**

Conversational agents are often deployed via mobile apps, desktop applications, web pages, or other Interfaces. You can use them to automate customer support, sales, marketing, education, entertainment, and other tasks. Some of them include the following:

**Customer service**

Businesses can use conversational agents to answer questions quickly and efficiently without hiring additional staff or paying agency fees to an outsourced call center. Additionally, chatbots can handle routine tasks such as resetting passwords or booking flights which is quite common in enterprise chatbots.

It can also answer basic product questions that don't require human judgment—making them ideal for low-cost services like online banking, where customers may not want to wait for a real person to get back to them.

**Information retrieval**

You can offer information about products or services through a chat interface instead of having the user search through articles on your website or in an app store search box.

The user can ask about the price of an item, for example, and you can provide that information in real-time. It can be anything from pointing a customer towards an item they're looking to purchase or providing details on how to use your product. It can also offer answers to more nuanced queries like how many days it'll take to receive a product, etc.

**Revenue optimization**

Conversational agents can optimize sales by suggesting products to customers who have not yet purchased them. They can collect first party data and be connected to customer relationship management (CRM) or email marketing software to send cart abandonment emails. It can also prompt users within the app/browser window to encourage a purchase.

It can help increase conversion rates by providing additional information about products that would go unnoticed. A conversational agent can replace pop up windows and act as a modern day concierge service for your site visitors.

**Examples of a conversational agent**

Here are a few examples of conversational agents that are currently being used in the market:

1. **Iris: Conversational agent for data science tasks**

While most conversational agents focus on simple tasks like offering customer support and booking appointments, Iris is coded differently. This agent can help users accomplish complex data science tasks like plotting a histogram from a dataset or conducting statistical analysis for those datasets.

The bot maps user inputs with pre-existing commands to decide what responses to offer. It does that by transforming those commands into automata that the bot can compose, sequence, and execute, providing the desired output. These commands require additional input, which the agent gets by either speaking to the user to resolve arguments or relying on previous conversations to understand the intended task.

*Iris's underlying model*

Using this agent, data scientists can complete predictive modeling tasks 2.6 times faster, decreasing the analysis time dramatically.

*Example of an executed Iris command*

2. **Woebot: Mental Health App**

Woebot is a mental health conversational agent that can help you monitor your mood and manage your mental health. It uses NLP, psychological expertise, and excellent copywriting to form a human-like conversation—making it easier for individuals to interact with it.

It works on the principles of Cognitive Behavior Therapy (CBT), a therapeutic approach to challenge recurring problematic thoughts. It can help anyone, irrespective of age, and a recent study confirmed its ability to reduce anxiety and depression in those who use it.

*Woebot's mental health app*

3. **Roof.ai: Real estate conversational AI chatbot**

Roof.ai is a conversational agent that helps real estate marketers automate interaction with leads and lead score assignment. Using Facebook as its prime channel, the bot interacts with potential leads and prompts them with questions that can help them qualify the lead. Once it assigns the score, it passes the conversation to a real estate agent who can take it forward.

*Roof.ai's real estate lead scoring app*

**Leverage conversational agents for business productivity**

With the current rise of conversational agents, businesses have better customer access, and operation costs have been significantly reduced. They have several use cases in different departments like logistics, marketing, customer support, etc.

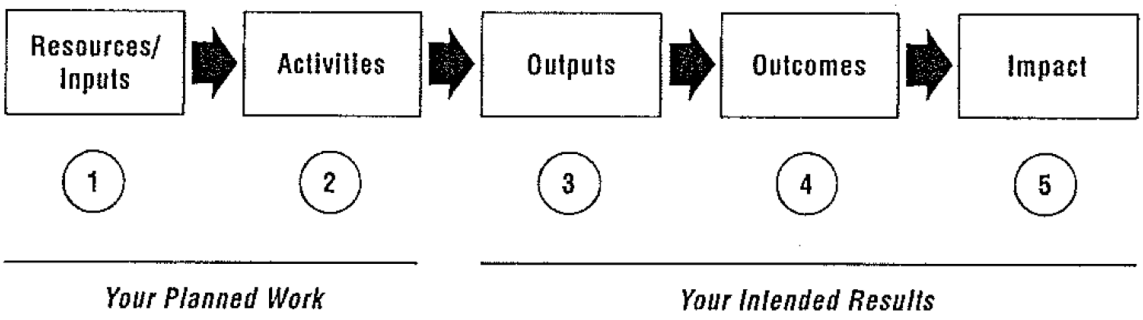
Although we still need human involvement, the agent can handle more challenging tasks and take care of more straightforward questions without involving a human operator.

It also helps build trust with customers and increases conversion rates. They also act as a shield between the user and human operator, reducing costs and allowing people more time to focus on other things while waiting for assistance.

As the technology progresses, conversational agents will become more accurate with time—increasing the pool of potential applications.

**Introduction to Logic Models**

Basically, a logic model is a systematic and visual way to present and share your understanding of the relationships among the resources you have to operate your program, the activities you plan, and the changes or results you hope to achieve.




The most basic logic model is a picture of how you believe your program will work. It uses words and/or pictures to describe the sequence of activities thought to bring about change and how these activities are linked to the results the program is expected to achieve.

**YOUR PLANNED WORK** describes what resources you think you need to implement your program and what you intend to do.

1. Resources include the human, financial, organizational, and community resources a program has available to direct toward doing the work. Sometimes this component is referred to as Inputs.

2. Program Activities are what the program does with the resources. Activities are the processes, tools, events, technology, and actions that are an intentional part of the program implementation. These interventions are used to bring about the intended program changes or results.

**YOUR INTENDED RESULTS** include all of the program's desired results (outputs, outcomes, and impact).

3. Outputs are the direct products of program activities and may include types, levels and targets of services to be delivered by the program.

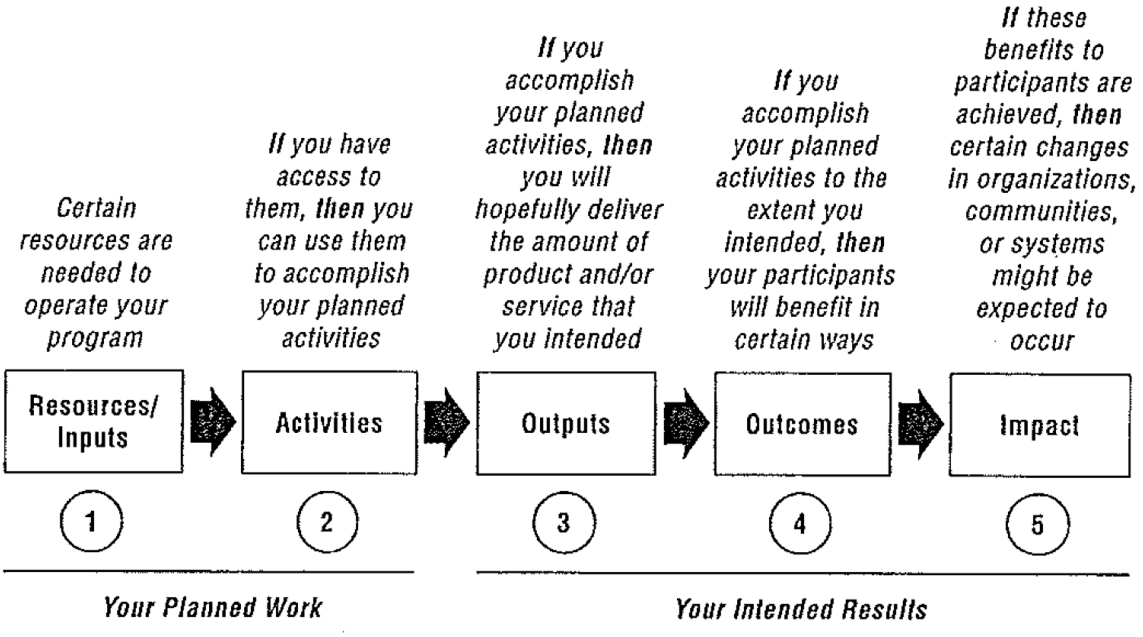
4. Outcomes are the specific changes in program participants' behavior, knowledge, skills, status and level of functioning. Short-term outcomes should be attainable within 1 to 3 years, while longer-term outcomes should be achievable within a 4 to 6year timeframe. The logical progression from short-term to long-term outcomes should be reflected in impact occurring within about 7 to 10 years.

5. Impact is the fundamental intended or unintended change occurring in organizations, communities or systems as a result of program activities within 7 to 10 years. In the current model of WKKF grant making and evaluation, impact often occurs after the conclusion of project funding.

The term logic model is frequently used interchangeably with the term program theory in the evaluation field. Logic models can alternatively be referred to as theory because they describe how a program works and to what end.

**How to "Read" a Logic Model**

When "read" from left to right, logic models describe program basics over time from planning through results. Reading a logic model means following the chain of reasoning or "If...then..." statements which connect the program's parts. The figure below shows how the basic logic model is read.




**Logic Model Purpose and Practical Application**

The purpose of a logic model is to provide stakeholders with a road map describing the sequence of related events connecting the need for the planned program with the program's desired results. Mapping a proposed program helps you visualize and understand how human and financial investments can contribute to achieving your intended program goals and can lead to program improvements. A logic model brings program concepts and dreams to life. It lets stakeholders try an idea on for size and apply theories to a model or picture of how the program would function. The following example shows how the logic model approach works.

**An Example:**

We are proposing an inexpensive family trip from Charleston, South Carolina, to Des Moines, Iowa, to visit relatives during December school holidays. The seasonal trip we dream of taking from Charleston to Des Moines is the "program." Basic assumptions about our trip "program" are:

• We want to visit relatives between 12/10/00 and 1/5/01 while the children are out of school.

• We will fly from South Carolina to Iowa because it takes less time than driving and because frequent flier (FF) miles are available.

• Using frequent flier miles will reduce travel costs.

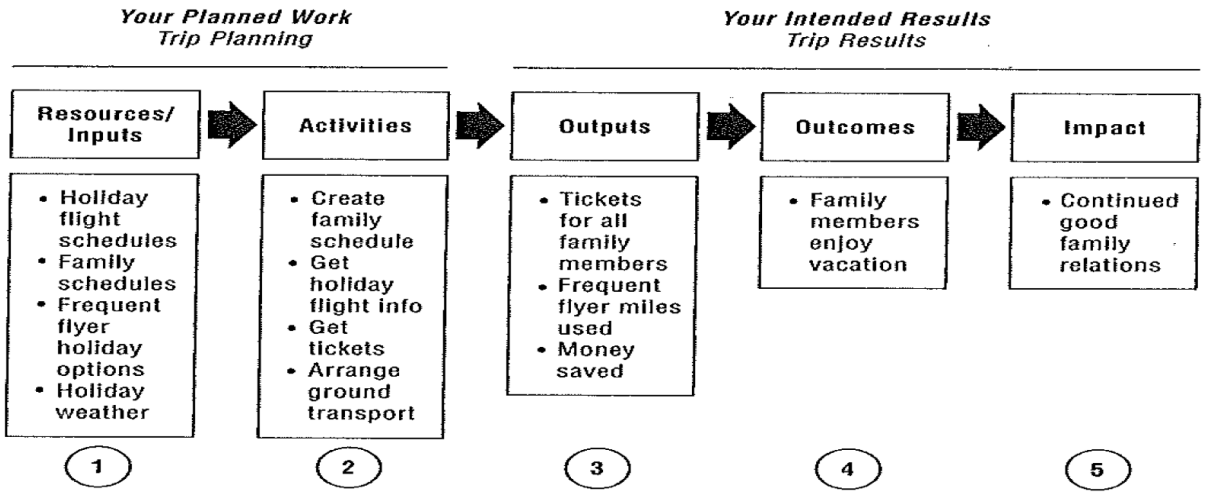
We have to determine the factors influencing our trip, including necessary resources, such as, the number of family members, scheduled vacation time, the number of frequent flier miles we have, round trip air reservations for each family member, and transportation to and from our home to the airport. The activities necessary to make this happen are the creation of our own family holiday schedule, securing our Iowa relative's schedule, garnering airline information and reservations and planning for transportation to and from the airport.

In this example, the results of our activities – or outputs – are mostly information, such as family schedules, flight schedules, and cost information based on the time frame of the trip. This information helps identify outcomes or

immediate goals. For instance, if we make reservations as soon as possible, we are able to find flights with available frequent flier slots and probably have more options for flights that fit within the time frame. Knowing this, our outcomes improve – reservations made well in advance result in flight schedules and airline costs that suit our timeline and travel budget. Longer-term impact of our trip is not an issue here, but might be projected as continued good family relationships in 2010.

Using a simple logic model as a trip-planning tool produced tangible benefits. It helped us gather information to influence our decisions about resources and allowed us to meet our stated goals. Applying this process consistently throughout our trip planning positions us for success by laying out the best course of action and giving us benchmarks for measuring progress – when we touch down in Charlotte and change planes for Cincinnati, we know we're on course for Des Moines.

Typical logic models use table and flow chart formats like those presented here to catalogue program factors, activities, and results and to illustrate a program's dimensions. Most use text and arrows or a graphic representation of program ideas. This is what our trip planning "program" could look like in logic model format.




It was easy to organize travel plans in a flow chart, but we could also choose to organize and display our thinking in other ways. A logic model does not have to be linear. It may appear as a simple image or concept map to describe more complex program concepts. Settling on a single image of a program is sometimes the most difficult step for program stakeholders.

**Theoretic Semantics:**

**Model-theoretic semantics** is a special form of truth-conditional semantics. According to it, the truth-values of sentences depend on certain abstract objects called models. Understood in this way, models are mathematical structures that provide the interpretations of the (non-logical) lexical expressions of a language and determine the truth-values of its (declarative) sentences. Originally designed for the semantic analysis of mathematical logic, model-theoretic semantics has become a standard tool in linguistic semantics, mostly through the impact of Richard Montague's seminal work on the analogy between formal and natural languages. As such, it is frequently (and loosely) identified with possible worlds semantics, which rests on an identification of sentence meanings with regions in Logical Space, the class of all possible worlds.

**Proof-theoretic semantics** is an alternative to truth-condition semantics. It is based on the fundamental assumption that the central notion in terms of which meanings are assigned to certain expressions of our language, in particular

to logical constants, is that of *proof* rather than *truth*. In this sense proof-theoretic semantics is *semantics in terms of proof*. Proof-theoretic semantics also means the *semantics of proofs*, i.e., the semantics of entities which describe how we arrive at certain assertions given certain assumptions. Both aspects of proof-theoretic semantics can be intertwined, i.e. the semantics of proofs is itself often given in terms of proofs.

Proof-theoretic semantics has several roots, the most specific one being Gentzen's remarks that the introduction rules in his calculus of natural deduction define the meanings of logical constants, while the elimination rules can be obtained as a consequence of this definition. More broadly, it belongs to what Prawitz called *general proof theory*. Even more broadly, it is part of the tradition according to which the meaning of a term should be explained by reference to the way it is *used* in our language.

**Symbolic computation** is concerned with algorithmic manipulations of symbolic objects. These can be objects in formal language, such as formulas or programs, or algebraic objects, such as polynomials or residue classes, or geometric objects, such as curves or surfaces. Research in symbolic computation combines advanced mathematics with advanced computer science for computing and development of algorithms. It is applied in various fields in science and engineering, such as the analysis of finite element methods, chemical reaction networks, wireless communication systems, statistical physics, robotics, and geometric modeling.

**What is a Symbolic Approach?**

When you were a child, you learned about the world around you through symbolism. With each new encounter, your mind created logical rules and informative relationships about the objects and concepts around you. The first time you came to an intersection, you learned to look both ways before crossing, establishing an associative relationship between cars and danger.

A symbolic approach works the same way, commonly used for NLP and natural language understanding (NLU), symbolic follows an IF-THEN logic structure. When an IF linguistic condition is met, the THEN output is generated. This makes it easy to establish clear and explainable rules, providing full transparency into how it works. In doing so, you essentially bypass the "black box" problem endemic to machine learning.

**What Are the Advantages of a Symbolic Approach?**

The flexibility to write your own rules is a significant advantage to using symbolic over machine learning for your NLP model, but that only scratches the surface of its benefits.

For instance, when machine learning alone is used to build an algorithm for NLP, any changes to your input data can result in model drift, forcing you to train and test your data once again. However, a symbolic approach to NLP allows you to easily adapt to and overcome model drift by identifying the issue and revising your rules, saving you valuable time and computational resources.

A symbolic approach also offers a higher level of accuracy out of the box by assigning a meaning to each word based on the context and embedded knowledge. This is process is called disambiguation and it a key component of the best NLP/NLU models.

A lack of language-based data can be problematic when you're trying to train a machine learning model. ML models require massive amounts of data just to get up and running, and this need is ongoing. With a symbolic approach, your ability to develop and refine rules remains consistent, allowing you to work with relatively small data sets.