CHAPTER 6

DISCOURSE PROCESSING

CHAPTER OVERVIEW

This chapter is concerned with discourse and world knowledge. A discourse comprises a sequence of sentences that must be interpreted with respect to the context. World knowledge is needed to find the connection between two sentences, to resolve ambiguities, and to infer new information to get a coherent view of the information being communicated through spoken or written text. Discourse and world knowledge is especially important for interpreting pronouns and the temporal aspects of the information conveyed, resolving ambiguities, and understanding metaphors and ellipses. The chapter introduces these concepts and discusses anaphor resolution methods. There are various patterns of discourse. Identifying the discourse relation between two or more sentences requires knowledge of how the discourse is structured. We, therefore, present a theoretical framework for discourse analysis.

6.1 INTRODUCTION

A discourse is most commonly described as the language above the sentence level or as 'language in use'. An arbitrary collection of sentences does not make sense to us. In order to make sense, a text must consists of sentences that are related to each other. This means that there exists a structure above the sentence that is needed for interpretation of text. This structure is known as the discourse structure, and the collection of interrelated sentences is a discourse.

However, Widdowson (1995) argued that a discourse unit can be smaller than a sentence. For instance, we easily interpret the word 'Ladies' written beside a seat on a bus. We do not use the dictionary definition of ladies to understand this. Instead, our interpretation is based on the context; it uses real world knowledge, which is not there in the text itself. Discourse analysis thus makes use of contextual knowledge.

Discourse analysis deals with the intended meaning of textual units. The following text illustrates this idea:

Excuse me. You are standing on my foot.

This sentence is not just a plain assertion; it is a request to someone to get off your foot. Again, the intended meaning is not present in the text itself. However, we understand the meaning because that is the way language is used generally.

The preceding discussion makes it clear what is meant by 'language above the sentence level' or 'language in use'. Discourse analysis involves the study of the relationship between language and contextual background. Contextual knowledge that is needed to interpret a sentence includes situational context, background knowledge, and co-textual context. Situational context is the knowledge about physical situations existing in the surroundings at the time of utterance. Background knowledge includes cultural knowledge and interpersonal knowledge. Co-textual context is the knowledge of what has been said earlier.

There are many types of discourse, including written, spoken, and signed discourse as well as monologue and dialogue. In this chapter, we focus on the monologue type of discourse. A monologue involves a speaker (writer) and a hearer (reader). The communication is unidirectional from speaker to hearer (whereas in a dialogue, the role of the participant alternates periodically). Many of the discourse problems are shared by the various types of discourse. However, they need different techniques to process them. Throughout this chapter, we use the term sentence and utterance interchangeably.

The phenomena that operate at discourse level include cohesion and coherence. Cohesion is a textual phenomenon, whereas coherence is a mental phenomenon. A text is cohesive if its elements link together, and coherent if it makes sense. Cohesion studies how words are linked to each other in the text. This linking can be forwards or backwards and meshes the text together. Language make use of cohesive devices like references, ellipsis, repetitions, and conjunctions to achieve this linking. We discuss these cohesive devices in Section 6.2. Resolving pronominal references is essential in applications like information extraction and text summarization. An information extraction system often needs to fill slots that correspond to named entities. In order to make this possible, pronouns referring to those named entities need to be identified first. Similarly, if a sentence in a text uses a pronoun that refers to an entity in a previous sentence, which is not included in the text, that sentence will be unreadable.

How we resolve these references is discussed in Section 6.3. In this chapter, we discuss these phenomena and the methods used to resolve them. What we speak or write appears unified. Coherence refers to this property of being meaningful and unified. We elaborate on this concept, explain various coherence relations, and discuss Hobbs's (1985) abductive framework for determining local coherence, in Section 6.4.

6.2 COHESION

Cohesion bounds text together. Consider the following piece of text:

> Yesterday, my friend invited me to her house. When I reached, my friend was preparing coffee. Her father was cleaning dishes. Her mother was busy writing a book.

Each occurrence of *her* in the preceding refers to the noun phrase *my friend's*. Here is the same text with *my friend's* substituted at each place where *her* appears.

> Yesterday, my friend invited me to my friend's house. When I reached, my friend was preparing coffee. My friend's father was cleaning dishes. My friend's mother was busy writing a book.

To most of us, this repetition is undesirable. This type of over-specification is avoided in the earlier text through the use of *her*. We say that *her* is cohesive with *my friend's*. This is just a simple example of cohesion with pronominal references. Any communication in a speaker-hearer environment assumes the presence of some shared knowledge. The encoding of this message by the speaker and its subsequent interpretation take place within the context of this shared knowledge. The speaker avoids encoding information that is obvious or known to the hearer.

Pronominal reference is just one type of reference. There are many others including ellipses, which we discuss next.

6.2.1 Reference

Reference is a means to link a referring expression to another referring expression in the surrounding text, as in the following example.

Suha bought a printer. It cost her Rs. 20,000. (6.1)

Here, 'her' refers to a person named 'Suha' and 'it' represents an entity named 'printer'. Both 'it' and 'her' in sentence (6.1) refer to entities that

have been previously introduced into the discourse. Such a reference is called an anaphoric reference. There are primarily five types of references: indefinite, definite, pronominal, demonstrative, and ordinal.

Indefinite Reference

An indefinite reference introduces a new object to the discourse context. The most commonly used form of indefinite reference involves the use of the determiner 'a' (or 'an') as in the following sentence.

I bought *a printer* today. (6.2)

Other markers of indefinite reference are the quantifier 'some' [sentence (6.3a)] and the determiner 'this' as in sentence (6.3b).

Some printers make noise while printing. (6.3a)

I met *this girl* earlier in a conference. (6.3b)

These references are also termed non-anaphoric references.

Definite Reference

A definite reference refers to an object that already exists in the discourse context, as illustrated in the following example.

I bought a printer today. The printer didn't work properly. (6.4)

The first sentence in this example introduces a new entity, which is referred to by the definite reference in the second sentence.

In most cases, the determiner structure of the noun phrase helps make the distinction between definite and indefinite referents.

Pronominal Reference

Pronominal references are the references that use a pronoun to refer to some entity. Sentence (6.5) illustrates this type of reference.

I bought a printer today. On installation, it didn't work properly. (6.5a)

Zuha forgets her pen drive in lab. (6.5b)

In (6.5a), 'it' refers to printer. In (6.5b), 'her' refers to Zuha.

However, the use of 'it' as a referent should be clearly distinguished from its pleonastic use, as in "It is raining" or in "It is okay", where 'it' refers to a state and not to an entity. Denber (1998) and Mitkov (1999) classify the use of 'it' as pleonastic in sentences like "It seems more likely that..." or "It is obvious that...". Though it appears that the phrase following 'that' can be regarded as a referent in these cases.

Pronominal referents put stricter constraints on the distance between the entities being referred to and their introduction. Usually, a pronominal reference refers to an object introduced in the previous one or two

sentences, whereas definite noun phrases can refer to objects further back.

A pronominal reference can refer to an entity before it is actually introduced in the discourse context, as in the following example.

Having installed it, I found that the printer was not working properly. (6.6)

This type of pronominal reference is called cataphoric reference.

A pronominal reference can be a part of a quantified context as in sentence (6.7).

All students should sign their project report. (6.7)

A reference need not always refer to a noun phrase. Sentence 6.8 illustrates a situation in which the pronominal referent 'it' refers to the event introduced in the previous sentence.

The printer I bought today doesn't work properly.
It surprised me. (6.8)

Here, 'it' refers to the event of printer not working properly.

Demonstrative Reference

I bought a printer today. I had bought one earlier in 2004. *This* one cost me Rs. 6,000 whereas that one cost me Rs. 12,000. (6.9)

Here, 'this' refers to the printer I bought today and 'that' refers to the printer I bought in 2004.

Quantifier or Ordinal Reference

An ordinal reference uses an ordinal, such as 'first', 'one', etc. Some authors call these referents as one-anaphora.

I visited a computer shop to buy a printer. I have seen many and now I have to select *one*. (6.10)

Here, 'one' refers to a printer I have seen.

Thus, it introduces a new entity into the discourse context. The use of 'one' in sentence (6.10) is different from the formal, non-specific use, as in the sentence, 'One should be confident while facing interviews', and its use as the numeral one, as in, 'She got her trousers shortened by one inch'.

Inferables

Inferable referents refer to entities that can be inferred from other entities explicitly evoked in the text, as illustrated in sentence (6.11).

I bought a printer today. On opening the package, I found *the paper tray* broken. (6.11)

Here, the paper tray does not introduce a new object in the discourse context; instead it refers to the paper tray of the printer introduced in first sentence.

Generic Reference

A generic reference refers to a whole class instead of an individual or specific entity.

> I saw two laser printers in a shop. *They* were the fastest printers available. (6.12)

Here, 'they' refers to laser printers in general and not to ones of those I saw.

6.2.2 Ellipsis

Ellipsis is a form of grammatical cohesion. It refers to the phenomenon where a part of a sentence (utterance) is omitted or left unpronounced. The reader uses the surrounding text to recover the omitted text. Ellipsis, like references, are cohesive devices that avoid repetition. Consider the following sentences:

> Do you take fish?
> Yes I do.

'Yes I do' is an example of an ellipsis in which the verb phrase has been deleted. Instead of saying 'Yes I take fish', the speaker omits the text 'take fish' because it is not necessary. The reader (hearer) retrieves the missing text from the previous sentence. Here is another example:

> I know that lady. Do you?

Here 'Do you' is an ellipsis, which stands for 'Do you know that lady'. Here, the verb phrase 'know that lady' is left out. The elided verb phrase is understood from the previous sentence.

The clause containing the elided verb phrase is called the target clause, and the clause from which the verb phrase is understood (taken) is called the source clause. The presence of references in the source clause may lead to ambiguities in target clause, as in the following example:

> Seema loves her mother and Suha does too.

This sentence may mean that Suha likes Seema's mother or that Suha likes her own mother. The first reading is often called a *strict* reading whereas the second reading is called a the *sloppy* reading. In a strict reading, the pronoun in the target clause refers to the same entity as the pronoun in the source clause.

6.2.3 Lexical Cohesion

References and ellipsis are forms of grammatical cohesion that avoid repetition of clauses. There are a number of other lexical phenomena that languages use to achieve cohesion, which come under the category of lexical cohesion. Unlike grammatical cohesion, lexical cohesion exploits repetition, either explicitly or implicitly, to introduce stylistic effect. Lexical cohesion devices include repetition, synonymy, and hypernymy. The use of repetition for style is illustrated in the following examples:

Ba ba black sheep,
Have you any wool?
Yes sir, yes sir,
Three bags full.

Instead of repeating the same word, synonymy uses a word that means the same in the given context; hypernymy uses a super-ordinate.

6.3 REFERENCE RESOLUTION

We now discuss the factors and methods that help resolve various types of referents.

6.3.1 Constraints and Preferences

Constraints that rule out certain referents need to be checked. This includes the constraints of number, gender, and case agreement. Apart from these strict constraints, there are a number of preferences and semantic constraints that can be used to identify a preferred referent. Let us have a look on these constraints and preferences.

Person Agreement

The referent and the referring expressions must agree in person. Consider the following examples:

Zuha and I bought a camera. *We* like capturing nature scenes.
(We = I and Zuha) (6.13a)

Zuha and Prabha bought a camera. *We* like capturing nature scenes.
(6.13b)

Resolving 'We' into 'Zuha and Prabha' is incorrect in (6.13b).

Case Agreement

The position where a pronoun is used constraints its form. For example, in the object position, we use the accusative case of a pronoun (e.g., him, her, them), whereas in the subject position, we use the nominative form of a pronoun.

Gender Agreement

English distinguishes between male, female, and non-personal genders in the use of third person pronouns. These constraints need to be checked when resolving pronominal references.

Zuha bought a printer. She is printing now.
(She = Zuha, not the printer) (6.14a)

Zuha bought a printer. It is printing now.
(It = the printer, not Zuha) (6.14b)

Selectional Restrictions

Selectional restriction, placed by verbs on their arguments, can also help in resolving references.

Zuha put an apple on the table. Suha is eating *it*.

The pronoun 'it' has two possible referents: apple and table. However, the verb eat requires some edible entity as its object. As the table is not edible, the correct referent is the apple.

As discussed in Chapter 5, violations to these restrictions are sometimes acceptable. That is why we usually call them preferences rather than restrictions. Quite often, these violations refer to metaphoric uses, as in the following sentence.

A range of shopping malls are opening in the city. The local vendors fear that *they* will swallow their livelihood.

Here, 'they' refers to shopping malls. Shopping malls are not something that can swallow, but its use is not semantically odd to us and is quite acceptable. Resolving these types of references requires semantic knowledge.

Recently Introduced References

While resolving references, the entities introduced more recently are considered of greater importance than those introduced further back in the text.

Grammatical Role

Grammatical roles played by an entity in a sentence, provide useful clues on their salience. For example, an entity in the subject position can be considered more important than one in the object position.

Parallelism

The structural parallelism that exists in a sentence can be used to resolve references, as in the following example:

Zuha went with Suha to the computer shop. Danish went with her to a computer institute.

The parallelism effect suggests that 'her' refers to Suha and not to Zuha.

Repeated Mention

This refers to the idea that entities that are focussed on in prior discourses are more likely to continue to be focussed on in subsequent discourses. Hence, it is more likely that they are referred to by the pronoun, as in the following excerpts from *SPAN* magazine:

Lucid was the first among the six women to join the astronaut program. A veteran of five space flights, logging 223 days in space, Lucid holds the international record for the most flight hours in orbit by any American, and any women in the world. She spent 180 days on the Russian space station Mir in 1996.

In 1998 ***she*** *wrote in* The Scientific American *that* ***she*** *viewed the Mir mission as the perfect opportunity to combine two of her passions: flying airplanes and working in laboratories.*

Here, the pronoun 'she' in the second paragraph refers to Lucid, who is the focus in the prior discourse.

Intra-sentential Syntactic Constraints

The reflexive use of pronouns is constrained by the syntactic relationship between the referential expression and the antecedent noun phrase. Usually, it co-refers with the subject of the innermost clause that includes it. The following examples clarify this constraint:

Preghna bought herself a laptop. [herself = Preghna] (6.15a)

Preghna bought her a laptop. [her ≠ Preghna] (6.15b)

6.3.2 Reference Resolution Algorithms

An algorithm that uses all these constraints and preferences requires a vast amount of knowledge, which is difficult to achieve. Till date, none of the existing algorithms for pronouns incorporates all of them. However, most use syntactic constraints, either directly or indirectly. In the following lines, we discuss pronoun resolution algorithms introduced by Lappin and Leass (1994), Grosz et al. (1995), Mitkov (1998), and Lappin (2003).

Resolution of Anaphora Procedure

Lappin and Leass (1994) proposed an algorithm that uses some of the constraints and preferences discussed in the previous section, to resolve pronominal anaphora. They called it RAP—Resolution of Anaphora Procedure. It uses a salience value, derived from the syntactic structure,

to rank a filtered set of NP candidates. It uses no semantic information. The algorithm uses a set of filters to identify pleonastic pronominal references and to eliminate candidate NPs that (i) do not agree in number, gender, or person, and (ii) violate syntactic co-reference constraints. The salience factors used by Lappin and Leass are listed in Table 6.1 along with their initial weights.

Table 6.1 Salience factors and their weights (Lappin and Leass 1994)

Sentence recency	100
Subject emphasis	80
Head noun emphasis	80
Existential emphasis	70
Accusative emphasis	50
Non-adverbial emphasis	50
Indirect object and oblique complement emphasis	40

The non-adverbial emphasis rewards NPs not occurring in the adverbial PPs. The salience factors considered by RAP, implements the following hierarchy of grammatical roles (adapted from Jurafsky and Martin 2000):

Subject > existential predicate nominal > direct object> indirect object | prepositional object > demarcated adverbial PP

The following are a few examples explaining these grammatical roles:

Suha bought a laser printer. (subject, head noun) (6.16a)

There are only *two printers* working properly in the lab. (existential predicate) (6.16b)

The engineer repaired the *printer*. (object) (6.16c)

Suha got her *printer* repaired today. (indirect object) (6.16d)

Suha placed the printer on the *table*. (prepositional object) (6.16e)

In *her new printer*, a scratch was found. (demarcated adverbial NP) (6.16f)

In sentence (6.16a), Suha receives 80 points for subject and 80 points for being denoted as a head noun. This is also true for Suha in sentences (6.16d) and (6.16e).

The salience value assigned to a candidate referent is the sum of the weights of salience factors associated with it. The referents in the current sentence are assigned initial weights as mentioned in Table 6.1. A sentence recency weight of 100 is assigned to all discourse referents introduced in the current sentence. Weights assigned by all salience factors to a referent before the current sentence are degraded by a factor of two. The steps in the pronoun resolution algorithm are as follows:

1. Create a list of potential referents.
2. Filter out potential referents that do not agree in number, person, and gender with the pronoun.
3. Remove referents that do not pass intra-sentential syntactic co-reference constraints.
4. Calculate the total salience value for each potential referent. Modify the salience value to account for role parallelism between the pronoun, the referent, and the cataphoric reference. In a cataphoric reference, a referent appears after the pronoun. Such a reference is penalized by assigning a negative weight of −175. Grammatical role parallelism refers to a situation in which the pronoun and the referring entity play the same role. A positive weight of 35 is assigned to such entities.
5. Select the referent with the highest salience value.

We now explain these ideas with the help of an example.

Example 6.1 Consider the following text:

Suha saw a laptop in the shop. She enquired about it.
She bought it. (6.17)

We begin with the first sentence and assign weights to the references, as shown in the following table:

	Recency	Subject	Existential	Object	Ind-obj	Non-adv	Head Noun	Total
Suha	100	80	—	—	—	50	80	310
Laptop	100	—	—	50	—	50	80	280
shop	100	—	—	—	—	50	80	230

The first sentence does not contain any pronominal reference. Now consider the next sentence. As mentioned earlier, we first degrade the salience value of each referent by a factor of two, as shown in the following table:

Referent	Phrases	Salience value
Suha	{Suha}	155
Laptop	{a laptop}	140
shop	{the shop}	115

The second sentence contains two pronouns: 'she' and 'it'. The gender agreement rules out 'laptop' and 'shop' from the list of potential referents for 'she' leaving only 'Suha'. So, the algorithm returns 'Suha' as the referent of 'she'. We now update the salience values by adding 'she' in the equivalence class for Suha and adding its salience score (=310) to Suha. Since 'she' is in the current sentence, it receives a recency score of

100. Other factors that contribute to its score are subject position (=80), not in adverbial position (=50), and head noun (=80), giving a total of 310. The updated values are listed in the following table:

Referent	Phrases	Salience value
Suha	{Suha, she}	465
Laptop	{a laptop}	140
shop	{the shop}	115

The next NP in the current sentence is 'it', which can refer to either 'shop' or 'laptop'. We first update the values by adding 35 to the salience score of 'laptop' to incorporate parallelism preference as both 'it' and 'laptop' are in the object position. As the weight of 'laptop' is more than that of 'shop', we select it as the referent.

The next table shows the updated discourse model after processing the second sentence. The score of 'it' is 100 (recency) + 50 (obj) + 50 (non-adv) + 80 (head-noun) = 280.

Referent	Phrases	Salience value
Suha	{Suha, she}	465
Laptop	{a laptop, it}	455
shop	{the shop}	115

We now move on to the last sentence. Before considering references in this sentence, we first reduce the values by half.

Referent	Phrases	Salience value
Suha	{Suha, she}	232.5
Laptop	{a laptop, it}	227.5
shop	{the shop}	57.5

The first noun phrase in the last sentence is 'she'. Step 2 of the algorithm suggests that the possible referent is Suha. So, we stop here and consider Suha as the referent. The following table shows the updated discourse model after this step. To distinguish 'she' in the current sentence from the one mentioned in first sentence, we represent it as she_1.

Referent	Phrases	Salience value
Suha	{Suha, she, she_1}	597.5
Laptop	{a laptop, it}	227.5
shop	{the shop}	57.5

The next noun phrase in the current sentence is 'it'. The possible referents are 'shop' and 'laptop'. Both 'it' and 'laptop' occur in the object position, so we add 35 for the parallelism preference to 'laptop', yielding a value of 245. As the salience score of laptop is high, 'it' will resolve to 'shop'.

Centering Algorithm

Centering theory (Grosz et al. 1995) uses a discourse model representation consisting of a forward-looking centre (C_f) and a backward-looking centre (C_b). The algorithm presumes the existence of a single entity 'centred' on a given point in the discourse. If U_{n-1} and U_n are two adjacent utterances, then the backward-looking centre of U_n, $C_b(U_n)$, represents the most prominent entity in the discourse after U_n has been interpreted. The forward-looking centre of U_n, $C_f(U_n)$, is an ordered list of entities mentioned in U_n. The centre $C_b(U_n)$, is the highest ranked entity of $C_f(U_{n-1})$. The elements of $C_f(U_n)$ are ordered based on their grammatical role in the utterance. The highest rank is given to a subject followed by the direct object, the indirect object, the oblique, and the adjuncts. This ranking is similar to that of the grammatical role hierarchy used by Lappin and Leass. However, no numerical weight is assigned to the entities on the list. The algorithm computes inter-sentential relationship between the pair of utterances U_{n-1} and U_n, as shown in Table 6.2, to find the preferred referents of the pronoun.

Table 6.2 Inter-sentential relationship

	$C_b(U_n) = C_b(U_{n-1})$ or undefined $C_b(U_{n-1})$	$C_b(U_n) \neq C_b(U_{n-1})$
$C_b(U_n) = C_p(U_n)$	Continuing	Smooth shift
$C_b(U_n) \neq C_p(U_n)$	Retaining	Rough shift

where $C_p(U_n)$ is the highest ranked, forward-looking centre in U_n, called preferred centre. The algorithm uses the following rules:

1. If some elements of $C_f(U_{n-1})$ are realized as the pronoun, then so is $C_b(U_n)$.
2. The transitions between utterances are ordered as follows:

   Continuing > Retaining > Smooth shift > Rough shift

The transition types define the referent assignment. The referent that results in the most preferred relation in Rule 2 is assigned to the pronoun. The algorithm uses the ordering of the element in the previous C_f list to break the tie. The steps in the algorithm are summarized as follows:

1. For each potential referent, generate $C_b - C_f$ combination.

2. Filter out candidate referents that violate the co-reference constraints (number, gender, person, selectional preferences, etc.).
3. Rank referents by transition ordering.

Example 6.3 Consider sentences (6.17), which are reproduced here.

Suha saw a laptop in the shop. She enquired about it. She bought it.

$C_f(U_1)$: {Suha, laptop, shop}
$C_p(U_1)$: Suha
$C_b(U_1)$: undefined

U_2 contains two pronouns: *she* and *it*. *She* is compatible with *Suha* and *it* is compatible with *laptop* and *shop*.

$C_b(U_2)$ = highest ranked element of $C_f(U_1)$ = {Suha}

It has two possible referents. Assuming that *it* refers to the laptop, the assignments would be

$C_f(U_2)$: {Suha, laptop}
$C_p(U_2)$: Suha
$C_b(U_2)$: Suha
Result: continue

Assuming that *it* refers to the shop, the assignments would be

$C_f(U_2)$: {Suha, shop}
$C_p(U_2)$: Suha
$C_b(U_2)$: Suha
Result: continue

Since both assignments result in a 'continue' transition, the ordering in $C_f(U_1)$ is used to break the tie. Since *laptop* precedes *shop* in C_f list of U_1, *laptop* is assigned as the referent.

In sentence U_3, *she* is compatible with Suha and *it* is compatible with *laptop*. Resolving them is trivial.

Mitkov's Pronoun Resolution Algorithm

Mitkov (1998) proposed a robust, knowledge-poor approach for resolving anaphors. His algorithm uses a noun phrase extractor and a filtering module to identify a list of potential candidates for the antecedent. The input to the algorithm is part-of-speech tagged text. The noun phrase extractor, extracts NPs up to two sentences back from the anaphor. A referential filter eliminates pleonastic (semantically null) uses of pronouns, as in 'It is important to note that...'. The filtering module filters out NPs that do not agree in number, person, or gender, with the pronoun. The remaining NPs constitute the list of potential referents. The algorithm

applies a list of antecedent indicators on them. These indicators are drawn empirically and implement syntactic salience or lexical preferences, some of which are domain dependent. The NP with the maximum aggregate score is proposed as the referent. To break the tie, a priority of preferences is used. The highest priority is given to candidate with higher score for immediate reference, followed by best collocation pattern score. If it does not work, the lexical (verb) preference score is used. If this also fails, the algorithm simply selects the most recent NPs from the candidate list as the referent. The antecedent indicators used by the algorithm are derived empirically. We now discuss these indicators.

Definiteness Definite NPs are more important for resolving anaphoric references than indefinite ones. The algorithm penalizes indefinite NPs by assigning them a negative score (–1). If no definite article, possessive or demonstrative pronouns, appear in the paragraph, the rule is ignored.

Givenness The NPs representing themes are considered good candidates for referents and are awarded a score of 1, the remaining NPs receive a score of zero. The theme appears first and has fair chances of having a co-referential link with the previous text.

Indicating Verbs Mitkov considered a set of verbs as good indicators for identifying salient NPs. The verb set considered by Mitkov is

{discuss, present, illustrate, identify, summarize, examine, describe, show, check, develop, review, report, outline, consider, investigate, explore, assess, analyse, synthesis, study, survey, deal, cover}

Mitkov considered the NPs following these verbs as preferred antecedents and assigned them a score of 1.

Lexical Repetition This indicator gives preference to lexically reiterated candidates. NPs that are repeated two or more times in a paragraph are assigned a score of 2. Sequences of NPs that are synonyms, or have the same head, are considered the same when counting their occurrences, e.g., printer, laser printer, and fast laser printer, are regarded the same.

Section Heading Preferences A candidate noun phrase that matches a noun phrase appearing in a section heading, gets a score of 1.

Non-prepositional Noun Phrases Non-prepositional noun phrases are preferred over noun phrases that are part of prepositional phrases. This preference is weakened by penalizing (–1) a prepositional noun phrase, for example,

Load the printer with paper before restarting it. [it = printer]

Paper is penalized for being part of the prepositional phrase 'with paper'. This helps resolve the pronoun 'it' correctly. NPs which are part of prepositional phrases are usually indirect object. This preference is thus in accordance with the following ranking used by the centering theory:

Subject > direct object > indirect object

Collocation Pattern Preference An NP whose collocation pattern is identical with that of a pronoun is assigned a weight of 2. The collocation pattern considered for this preference is NP|Pronoun Verb and Verb NP|Pronoun.

Immediate Reference The heuristic used to implement this preference is described by the following pattern. (You) $Verb_1$ NP...(con) (you) $Verb_2$ it (con (you) $Verb_3$ it), where con $\in$ { and /or / before/ after/...}. The noun phrase immediately after $Verb_1$ is a more likely candidate for the antecedent of the pronoun 'it' immediately following $verb_2$ and is therefore given a preference by assigning it a score of 2.

Referential Distance The referential distance indicator suggests that noun phrases occurring recently are more likely candidates than those introduced further back. This preference is realized in a complex sentence using the following preference hierarchy.

> Noun phrases in the previous clause (score = 2) > noun phrase in the previous sentences (score =1) > noun phrases occurring two sentences back (score = 0) > noun phrases occurring three sentences back (score = -1).

For a simple sentence, noun phrases that occurred in the previous sentence are the best candidates for referents, followed by noun phrases situated two sentences back, followed by noun phrases situated three sentences further back (score 1, 0, –1 respectively).

Term Preference NPs representing terms in the field are deemed better than non-terms, and are therefore assigned a score of 1.

Mitkov pointed out that the antecedent indicators are just preferences, not absolute factors. In certain cases, one or more indicators do not point to the correct antecedent. Sometimes, the erroneous clues given by some indicators are counterbalanced by preferences assigned by other indicators. The steps in the algorithm are summarized in Figure 6.1. To give a concrete idea, we illustrate the algorithm with an example.

1. Extract noun phrases in the current sentence and previous two sentences (if available). Consider only the noun phrases appearing to the left of the anaphora (pleonastic references are dropped).
2. Filter out NPs which do not agree in number, gender, or person, The remaining NPs are potential candidates.
3. Apply antecedent indicators to each potential candidate. The candidate with the maximum aggregate score is accepted as the referent of the pronoun. Break the tie using the collocation pattern score. If this does not work, use the lexical (verb) preference score, otherwise select the most recent NPs from the candidate list as the referent.

Figure 6.1 Steps in Mitkov's pronoun resolution algorithm (adapted from Mitkov 1998)

Example 6.2 Consider sentences (6.17), which are reproduced here.

Suha saw a laptop in the shop. She enquired about it. She bought it.

Noun phrases in the first sentence are: {Suha, laptop, shop}.

Now consider the pronoun 'she' in the second sentence. As mentioned in Step 1 of the algorithm, the potential candidates are NPs to the left of anaphor, i.e., NPs identified in sentence 1. Step 2 eliminates 'laptop' and 'shop' from this list, due to non-agreement in the gender, leaving only one potential candidate. Hence, we accept 'Suha' as the antecedent of 'she'. Next, the pronoun to be resolved is 'it'. The candidate referents for 'it' are {Suha, laptop, shop}. Step 2 eliminates Suha from this list. Now, we apply the antecedent indicators to each of the remaining candidates. As 'shop' is a prepositional NP, a negative score (–1) is assigned to it. The total score for shop is –1 + definiteness 1 + referential distance 1 + indicating verbs 0 + term preference 1 + section heading 0 + collocation 0 = 2; and for 'laptop' the score is definiteness 1 + referential distance 1 + indicating verbs 0 + term preference 1 + non-prepositional noun phrase 0 + section heading 0 + collocation 0+ givenness 1 = 4. So, we correctly resolve 'it' into laptop. Identifying the antecedent of pronouns in the third sentence is left as an exercise to readers.

Mitkov reported a success rate of 89.7% for the genre of technical manuals, which is better than any other success rate existing on the same genre. In particular, any knowledge-poor algorithm will have difficulties with sentences that have more complex syntactic structures. This is because the algorithm does not use any syntactic knowledge. The approach can be easily adapted for other languages as well.

Sequenced Model

Lappin (2003) proposed an integrated sequenced model of anaphora and ellipsis resolution. Figure 6.2 outlines the architecture of this model. The

sequenced model combines relatively inexpensive and robust syntactic salience and recency-based approaches, with lexical preference model and adductive inference model, to achieve higher accuracy. These models are invoked one after the other in sequence. First, the syntactic salience and recency-based methods are applied to identify referring expressions for a pronoun. If this method fails to reliably resolve some references, then the lexical preference model is invoked. The unresolved cases include references for which the difference in the salience scores of top two candidates is less than a certain threshold value. The lexical preference model exploits semantic and real-world knowledge to resolve these references. The computational cost incurred in this approach is higher than in the first model. If all the references are resolved successfully, the process terminates, otherwise an even more expensive abductive inference model (Kehler 2000, 2002) is called.

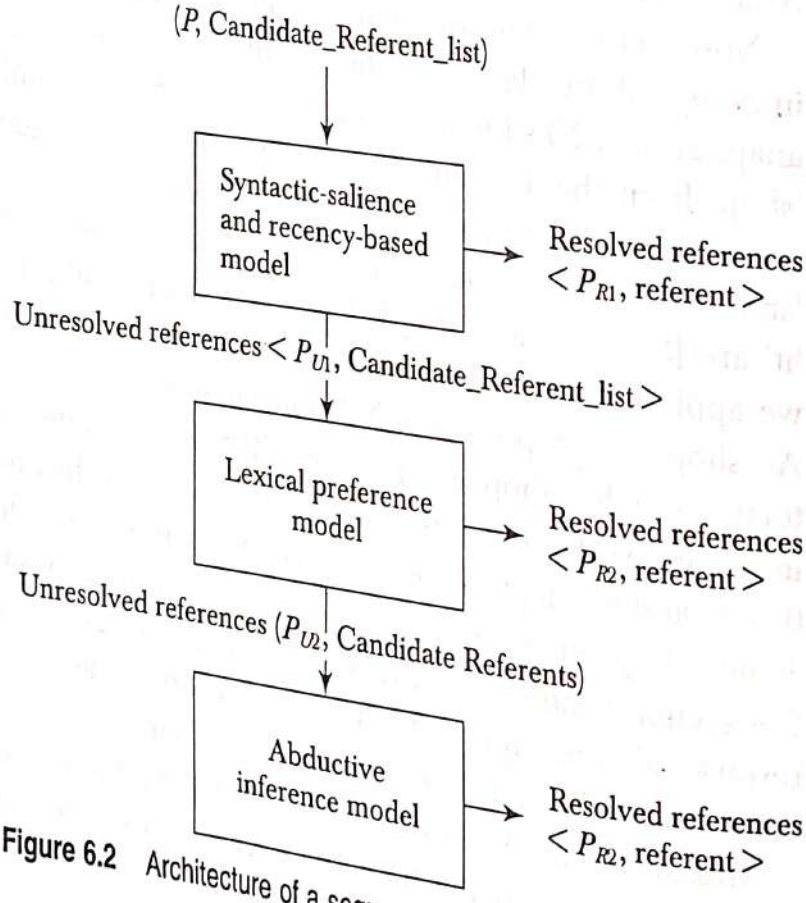

Figure 6.2 Architecture of a sequenced model

6.4 DISCOURSE COHERENCE AND STRUCTURE

Consider the following text:

Biomass is emerging as a viable source of power for rural electrification in India. At first glance, Kirgavalu may look like a typical village in southern Karnataka.

Both sentences in the passage are well formed and independently interpretable. But the passage seems a bit odd. The reason is that we try to establish a connection between the first and the second sentence. We raise questions such as how the Kirgavalu village is connected with biomass. In this case, we find it difficult to understand the connection. By raising such questions, we point out that the discourse is not coherent. In order to make the text coherent, we might build an understanding that perhaps the Kirguvalu village has a biomass plant. The attempt on the part of hearer (reader) to establish a connection between a pair of sentences suggests that merely grouping well-formed, independently interpretable sentences, does not yield meaningful passages; coherence is required to produce a meaningful composition. It is also needed for discourse comprehension. Coherence is different from cohesion. *Cohesion* refers to the grammatical relationship between words, referring forwards or backwards to other words, or substituting words or phrases, within the text.

There are a number of relations that connect utterances (sentences). Consider the following text:

Section 5.2 deals with sentence level meaning representation. (6.18a)

In particular, we discuss the general characteristics of meaning representation languages (Section 5.2.1), (6.18b′)
and computational approaches to semantic analysis (syntax-driven semantic analysis and semantic grammars in Section 5.2.3). (6.18b″)

Next, we discuss the internal structure of words, their relationships, and their meanings in Section 5.3. (6.18c)

Sentence (6.18a) introduces a topic. The next sentence elaborate on that by breaking it into subtopics, clauses (6.18b′) and (6.18b″). A temporal relation exists between (6.18b) and (6.18c) indicated by 'next', which links the topics we intend to discuss. Figure 6.3 illustrates these relations.

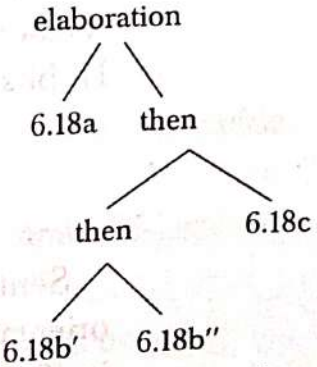

Figure 6.3 Structure of passage (6.18)

A number of researchers have pointed out that such relations exist and have proposed various instances. Joseph Grimes in *Thread of Discourse* (1975) includes alternation, specification, equivalence, attributions, and explanations. Grimes called these relations rhetorical predicates,

whereas Hobbs called them coherence relations. Robert Longacre (1976) include conjunction, contrast, comparison, alternation, temporal overlap and succession, implications, and causation. The list of coherence relations proposed by Hobbs (1979) includes result, explanation, occasion, parallel, and elaboration.

6.4.1 Coherence Relations

Hobbs (1985) described the process of interpreting discourse as 'a process of using knowledge acquired in past to construct a theory of what is happening in the present.' Understanding discourse requires identification of the coherence relations in the discourse. We now illustrate some coherence relations. In all of these examples, we assume that S_0 and S_1 are two consecutive utterances (sentences). Cue phrases, such as 'so', 'because of', 'but', 'while', and 'therefore', are good indicators of relations between discourse segments.

Occasion

Consider the following passage:

> At 9:00 a.m., the train arrived at Allahabad. The conference was inaugurated at 10 a.m. (6.19)

One way to read the passage to make it coherent is by assuming that someone who wants to attend the conference was on the train. That is, the first event sets up the occasion for the second. This relation is different from the causality relation. There is nothing special about the train that causes the conference to be inaugurated.

There are two cases that define occasion relation:

1. A change of state can be inferred from the assertion of S_0, whose final state can be inferred from S_1.
2. A change of state can be inferred from the assertion of S_1, whose initial state can be inferred from S_0.

Cause and enablement can be regarded as special cases of the occasion relation. Here is another illustration of occasion relation (adapted from Hobbs 1985):

> Go out of this door. (6.20a)
> Turn right. (6.20b)
> Go to the second room. (6.20c)

Sentence (6.20a) describes a change of location and assumes an orientation. The final state of the location holds during the event described in (6.20b). The initial state of the change in location described in (6.20c) can be inferred from (6.20b). Similarly, the orientation assumed in (6.20a)

is the initial state for the change in state described in 6.20b, and its final state is assumed in 6.20c. Figure 6.4 shows the inferences that need to be drawn to satisfy the definition. It is possible to find more than one relation between a pair of sentences, provided they do not involve inconsistent assumptions.

Type	6.20a	6.20b	6.20c
1	loc 1 → loc 2	loc2	
2		loc 2	loc 2 → loc 3
2	angle 1	angle 1 → angle 2	
1		angle 1 → angle 2	angle 2

Figure 6.4 Occasion relation in sentences (6.20)

Here is another example of occasion relation:

Increment the counter by one.
If it is 100, reset it to zero. (6.21)

Here, the value of the counter is changed, which is presupposed in the second sentence.

Explanation

The segment S_1 is an explanation of S_0 if S_1 describes an event or state that could cause the state or event described in S_0. Explanation is a relation that relates a segment of discourse to the listener's prior knowledge. It is formally defined as follows:

Infer that the state or event asserted by S_1 causes, or could cause, the state or event asserted by S_0.

Suha ate all the rice in bowl. She was very hungry. (6.22)

In this passage, S_1 explains the event asserted by S_0.

Causality may sometimes be explicitly stated as in the following statements:

Suha ate all the rice in the bowl because she was very hungry.
I get late because of the procession on the roads.

Elaboration

The definition of the elaboration relation involves identical entities. It is defined formally as follows:

Infer the same proposition from the assertions of S_0 and S_1.

A simple example of elaboration relation is

Saif scored an unbeaten century today. He was in full swing and made 108 not out on 87 balls. (6.23)

From the first sentence, and from what we know about an unbeaten century, we infer that Saif made more than 100 runs and he remains not out. By assuming that 'he' refers to Saif, we infer the same proposition from S_1 and thus establish the elaboration relation.

Parallel

The parallel relation is based on the similarity of entities. In this context, we say that two entities are similar if they share some property. More formally, the parallel relation is defined as follows:

Infer $p(x_1, x_2,...)$ from the assertion of S_0 and $p(y_1, y_2, ...)$ from the assertion of S_1, where x_1 and y_2 are similar, for all i.

Suha likes reading novels. Zuha enjoys reading science fiction. (6.24)

For each of the segments, we infer that a person likes reading books. Zuha and Suha are similar in that they are both people, reading novels, and reading science fiction. The predicate in this case may be hobby.

Contrast

There are two cases that define contrast relations:

1. Infer $p(x)$ from the assertion of S_0 and $\neg\, p(y)$ from the assertion of S_1, where x and y are similar.
2. Infer $p(x)$ from the assertion of S_0 and $\neg\, p(y)$ from the assertion of S_1, where there is some property q such that $q(x)$ and $\neg\, q(y)$.

Here is a simple example of the first case:

Suha does not like cricket. But she likes cricket more than any other game. (6.25)

Exemplification

Infer $p(X)$ from the assertion of S_0 and infer $p(x)$ from the assertion of S_1, where x is a member or subset of X. This relation is illustrated as follows:

Suha bought a printer today. It is a laser printer. (6.26)

6.4.2 Discourse Interpretation

Hobbs (1985) suggested that the problem of discourse interpretation can be solved by decomposing it into six sub-problems.

1. Logical Notation or Knowledge Representation

The first sub-problem deals with the problem of representation. In order to interpret discourse, a logical representation of natural language sentences is required. First order predicate logic representation is one such representation that has been used to translate natural language

representation into logical representation, and supports reasoning based on that representation.

2. Syntax and Semantics

This is concerned with the translation of text, sentence by sentence, into logical notation or representation. This problem has been researched heavily in linguistics and computational linguistics (Woods 1970; Montague 1974), and is considered to be solved to a large extent for common syntactic constructions.

3. Knowledge Encoding

This deals with the representation of the world and the language in the knowledge base. This knowledge is required for interpreting discourse. However, the task of encoding world knowledge is not trivial. We must decide what knowledge to represent, how to represent it, and whether the new knowledge being added is consistent with what is already in existence. A lot of research is focused on this problem. We make some general assumptions about how knowledge is encoded and assume the existence of specific facts in the knowledge base, so that we may continue to the discussion of the more important problem of 'how to use this knowledge in interpretation'. For example, we can have the following fact in knowledge base:

$$(\forall x)(\exists y)\ \text{printer}\ (x) \rightarrow \text{cartridge}\ (y, x)$$

4. Deductive Mechanism

In order to use stored facts, we must have some deductive mechanism. One such rule of inference is modus ponens, which permits us to infer cartridge (y, x) from:

$$(\exists x)\ \text{printer}\ (x)$$

and $(\forall x)\ (\exists y)\ \text{printer}\ (x) \rightarrow \text{cartridge}\ (y, x)$

5. Discourse Operations or Specifications of Possible Interpretation

There are certain problems in discourse such as co-reference resolution. These problems need to be resolved first to interpret text. This requires the identification of these problems and a specification of what it means to solve them. For example, a specification might state that the existence of an entity described by the definite noun phrase, can be inferred from the previous text and knowledge base.

6. Specification of the Best Interpretation

Discourse operations may yield many solutions to a discourse problem. This sub-theory deals with identifying the most economic interpretation for a sentence. The factors that govern cost of the solution include

complexity of proof, salience of axiom used, and redundancy in the interpretation. Let us take a close look at what the discourse problems are. Discourse problems can be divided into those problems that can be solved using information within the sentence, and those that involve the relation of the sentence to its context.

The within-sentence problems include problems of co-reference resolution, e.g., resolving pronouns, definite noun phrases, and missing arguments; identifying intended predicates where predicate is non-specific; satisfaction of selectional constraints; and determining the internal coherence of the discourse. The internal coherence problem deals with inferring relationship, such as causality, between sentences.

The second type of discourse problem considers the relationship between the sentence and the world.

6.4.3 Abductive Interpretation of Local Coherence

Hobbs (1985) presented an abductive framework for determining local coherence of the utterance. 'Abductive' means that assumptions are allowed at various costs. The method seeks the most economic interpretation of a sentence, such as an explanation that uses a small number of assumptions or one that uses the most specific properties of the input. In abductive inference, we make assumptions that need not be provable. Hobbs used an etc. predicate to represent all other properties that must be true for an axiom, but which were too vague to be stated explicitly. These predicates are assumed at a certain cost, not proved. A predicate with a low assumption cost will be preferred to one with high assumption cost. We now explain how the coherence of a segment is established with the help of an example.

Example 6.3 Consider the following text:

The local administration stopped the trade union from meeting. They feared violence. (6.27)

We need to establish local coherence in this segment. One way to prove that there is a coherence relation between the sentences is to prove that there is an explanation relation, i.e.,

Explanation ($e1$,$e2$)

This relation will hold if there is a causal relation between them:

Cause ($e2$,$e1$)

The logical form of the sentences and the hypothesized causal relation between them is given by the following expression.

$(\exists\ s, l, m, u, f, y, v)$ stops $(s, l, m) \wedge$ meeting $(m, u) \wedge$ cuase $(f, s) \wedge$ fear $(f, y, v) \wedge$ violent (v, z)

To prove this expression, we require axioms representing world knowledge in addition to axioms about coherence relation. The world knowledge needed for establishing coherence in this example is as follows:

1. If there is a fear event f imposed by someone, say y, of violence v, it means that y does not want violence (v).
2. A meeting m, by trade union u, causes violence.
3. If someone y, does not want (diswant) event v, and v is caused by m, then that will also cause y to diswant m.
4. If the local administration does not want something, then they will stop it.
5. And finally, cause is transitive, i.e., if $e1$ causes $e2$ and $e2$ causes $e3$, then $e1$ causes $e3$.

The axioms representing these sentences are as follows:

$(\forall f, y, v)$ fear $(f, y, v) \rightarrow \exists\ d)$ diswant $(d, y, v) \wedge$ cause (f, d)
$(\forall m, u)$ meeting $(m, u) \rightarrow (\exists\ v, z)$ cause $(m, v) \wedge$ violent (v, z)
$(\forall m, v, d, y)$ cause $(m, v) \wedge$ diswant $(d, y, v) \rightarrow (\exists\ d1)$ diswant $(d1, y, m) \wedge$ cause $(d, d1)$
$(\forall d1, l, m)$ diswant $(d1, l, m) \wedge$ localadministration $(l) \rightarrow (\exists s)$ stop $(s, l, m) \wedge$ cause $(d1, s)$
$(\forall e1, e2, e3)$ cause $(e1, e2) \wedge$ cause $(e2, e3) \rightarrow$ cause $(e1, e3)$

The derivation is shown in Figure 6.5. During the derivation, we also unify y with l.

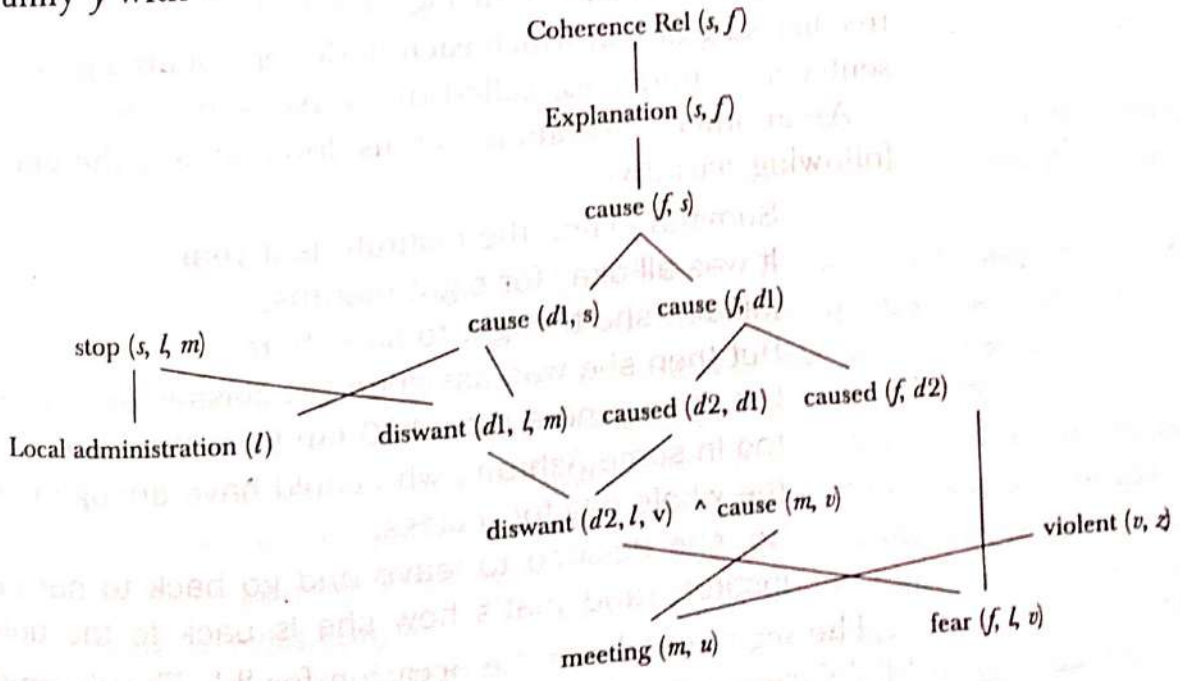

Figure 6.5 Interpretation of sentences (6.27)

Sometimes, the speaker uses a connective, which makes coherent relations explicit. For example, the use of 'because' to join the two sentences in segment (6.27), resulting in *The local administration stopped trade union from meeting because they feared violence,* would make the coherence relation explicit. The literal cause (f, s) becomes part of the logical form of the sentence and we need not to assume it.

We can extend this framework to establish the coherence of larger discourse.

6.4.4 Discourse Structure

So far we have discussed only the relations between a pair of sentences. In fact, it is possible to establish such relations in longer discourse. A discourse has a structure. For example, sentences (6.18b′) and (6.18b″) are related by an occasion relation. They combine to give a segment, which is linked with (6.18c) by an occasion relation. The resulting composed segment is related to (6.18a) by an elaboration relation. The coherence relationships between these sentences assign a 'coherence structure' to the discourse, as shown in Figure 6.6. It is a tree-like structure in which each node represents a group of locally coherent sentences (utterances) called discourse segments.

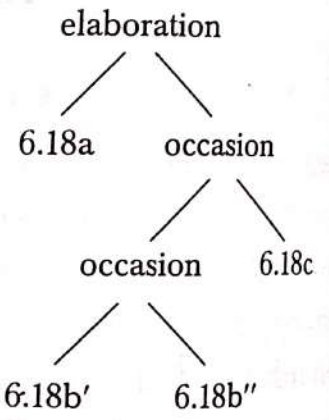

Figure 6.6 Coherence structure of text (6.18)

As another illustration, let us look at a coherence structure of the following narrative:

Sumitha joined the institute last year. (6.28a)
It was all okay for eight months, (6.28b)
Initially, she thought to stay there. (6.28c)
But then she was assigned UG classes at a remote centre. (6.28d)
Um, it was more than 100 km from the institute and that too in some Ashram, who could have enough time to waste the whole day for a class.
So, she decided to leave and go back to her parent (6.28e)
institute | and that's how she is back to the university. (6.28f) (6.28g)

The segment 'a' sets the occasion for 'b'. The circumstance of segment 'd'–'e' causes and thus occasions the events of 'f'. The segments 'c' and 'f' are contrasting relations. 'a'–'e' and 'f' are related by a set of events and its outcome. Figure 6.7 illustrates the structure.

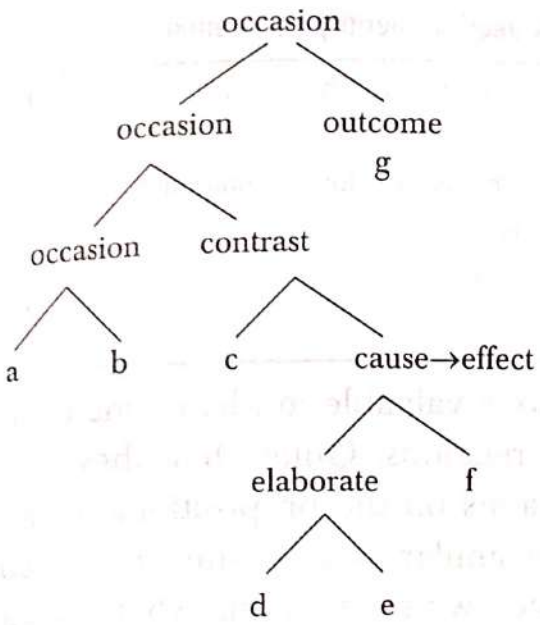

Figure 6.7 Structure of text (6.28)

Such a structure of discourse is useful in explaining classical problems of 'topic', 'genre', and 'coherence drift' that occur in ordinary conversation.

Hobbs proposed a four-step procedure for analysing discourse. As we move from one step to another, difficulty increases. The steps involved in the procedure are discussed below.

1. Identify one or two major breaks in the text and divide the text into two or three segments. This division corresponds to most natural division that one can carry out intuitively. This process is repeated for each segments obtained at first place. The iteration continues until we get single clause. The output of this step is a tree structure of the text. For example, in text (6.18), the major break comes between sentences (6.18a) and (6.18c). Within sentence (6.19b), there is a break between the first clause and the second clause of the sentence. This gives the tree structure of Figure 6.5.
2. In step 2, the non-terminal nodes of the tree are labelled with coherence relations. We follow a bottom-up approach to get an understanding of what is being representing by the composed segment. Thus, in Figure 6.5, the node linking (6.18b′) and (6.18b″) is labelled with the occasion relation. The node linking the resulting segment and (6.18c) is labelled with occasion relation and son. This step requires an understanding of different types of relations. Some simple heuristics based on what conjunctions need to be inserted might help identifying coherence relation. For example, if we can insert 'then' between S0 and S1 and reversing the order of the segment changes the sense, then the occasion relation is quite likely. If 'because' seems to be appropriate between S0 and S1, then explanation relation is preferred candidate. Table 6.3 lists useful conjuncts to identify coherence relation.

Table 6.3 Conjunctions used to identify coherence relation

Explanation: because, and so, hence, That's why
Occasion: then
Elaboration: Also, i.e. or that is, in addition, note that
Parallel: Similarly, likewise
Exemplification: for example
Contrast: but, however

These indicators prove valuable in identifying coherence relation b they do not define relations. Quite often they work because usuall they impose constraints on the propositional content of clauses the link which are quite similar to those imposed by coherence relations

3. This step is concerned with identifying what knowledge underlies the (discourse) the composed segment. We need to specify the knowledge or beliefs that support assignment of coherence relation to the nodes.
4. Step 4 is concerned with validation of hypotheses made in step 3. This requires consideration of longer corpus and construction of a knowledge base that would support the analysis of al of the text in the corpus.

SUMMARY

This chapter introduces discourse concepts and presents methods for analysing discourse.

- Discourse is concerned with linguistic and extra-linguistic phenomena that give text a unified and meaningful form.
- Discourse can be defined as 'language above the sentence level.
- Another way, we can describe discourse is by saying that it is 'language in use'.
- Discourse processing uses situational context, and cultural, social, interpersonal, and linguistic knowledge.
- Language uses cohesive devices like references, ellipsis, conjunctions, repetitions, etc., to achieve bound text.
- Ellipsis refers to the phenomenon where part of a sentence (utterances) is omitted or left unpronounced.
- Reference resolution techniques use various constraints and preferences to identify the preferred referent.
- Cohesion relates words to other words within the text; coherence relates sentences.
- The coherence relationships between sentences assign a 'coherence structure' to text.

REFERENCES

Brennan, S., M. Friedman, and C. Pollard, 1987, 'A centreing approach to pronouns,' *ACL-87*, pp. 155–62.

Denber, Michel, 1998, 'Automatic resolution of anaphora in English,' Technical report, Eastman Kodak Co., http://www.wlv.ac.uk/~le1825/anaphora_resolutionpapers/denber.ps.

Grimes, Joseph, 1975, *The Thread of Discourse*, Mouton and Company, The Hague, The Netherlands.

Grosz, B. J., A.K. Joshi, and S. Weinstein, 1995, 'Centering: a framework for modelling the local coherence of discourse,' *Computational Linguistics*, 21(2), pp. 175–204.

Hobbs, Jerry, 1979, 'Coherence and coreference,' *Cognitive Science*, 3, pp. 67–90.

_____1985, 'On the coherence and structure of discourse,' Technical Report 85-37, Center for the Study of Language and Information (CSLI), Stanford, CA.

Kehler, A., 2000, 'Pragmatics,' Chapter 18 in D. Jurafsky and J. Martin (Eds.), *Speech and Language Processing*, Prentice Hall, Upper Saddle River, NJ.

_____2002, *Coherence, Reference, and the Theory of Grammar*, Stanford, CSLI, CA.

Lappin, S., 2003, A 'sequenced model of anaphora and ellipsis resolution,' *Anaphora Processing: Linguistic, Cognitive, and Computational Modelling*, A. Branco, A., McEnery, and R. Mitkov (Eds.), John Benjamins, Amsterdam, pp. 3–16.

Lappin, Shalom and Herbert J. Leass, 1994, 'An algorithm for pronominal anaphora resolution,' *Computational Linguistics*, 20(4), pp. 535–61.

Longacre, Robert, 1976, *An Anatomy of Speech Notions*, The Peter Rider Press, Ghent, Belgium.

Mann, William and Sandra Thompson, 1986, 'Relational propositions in discourse,' *Discourse processes*, 9(1), pp. 57–90.

Mitkov, R., 1998, 'Robust pronoun resolution with limited knowledge,' *Proceedings of the 18th International Conference on Computational Linguistics (COLING'98)/ACL'98 Conference*, Montreal, Canada, pp. 869–75.

_____1999, 'Anaphora resolution: The state of the art, 1999,' Paper based on the COLING'98/ACL'98 tutorial on anaphora resolution, University of Wolverhampton.

Widdowson, H.G., 1995, 'Discourse analysis—a critical view,' *Language and Literature*, 4(3), pp. 157–72.

CHAPTER 12
LEXICAL RESOURCES

CHAPTER OVERVIEW

This chapter introduces various tools and lexical resources used in text processing applications and provides a ready reference of these. In particular, it introduces the reader with tools such as stemmers and taggers, lexical resources such as WordNet and FrameNet, and test collections (corpuses) that are freely available for research purpose.

12.1 INTRODUCTION

A whole range of tools and lexical resources have been developed to ease the task of researchers working with natural language processing (NLP). Many of these are open sources, i.e., readers can download them off the Internet. This chapter introduces some of the freely available resources. The motivation behind including this chapter comes from the belief that *knowing where the information is, is half of the information.*

We hope that providing a ready reference of what is available where, will save a lot of time and effort, especially for young researchers and those who are new to the field. All the material presented in this chapter is already available, at the links provided with the discussion or in the form of scholarly articles published on that resource. We bring these resources together and offer a brief discussion on them. In particular, we discuss lexical resources such as WordNet and FrameNet, and tools such as stemmers, taggers, and parsers, and freely available test corpuses for various text-processing applications. We begin our discussion with WordNet in Section 12.2. Section 12.3 discusses FrameNet. Stemmers are discussed in Section 12.4. We present a list of available part-of-speech taggers in Section 12.5. The next section presents a list of document collections. And finally, relevant journals and conferences are listed in Section 12.7.

12.2 WORDNET

WordNet[1] (Miller 1990, 1995) is a large lexical database for the English language. Inspired by psycholinguistic theories, it was developed and is being maintained at the Cognitive Science Laboratory, Princeton University, under the direction of George A. Miller. WordNet consists of three databases—one for nouns, one for verbs, and one for both adjectives and adverbs. Information is organized into sets of synonymous words called *synsets*, each representing one base concept. The synsets are linked to each other by means of lexical and semantic relations. Lexical relations occur between word-forms (i.e., senses) and semantic relations between word meanings. These relations include synonymy, hypernymy/hyponymy, antonymy, meronymy/holonymy, troponymy, etc. A word may appear in more than one synset and in more than one part-of-speech. The meaning of a word is called sense. WordNet lists all senses of a word, each sense belonging to a different synset. WordNet's sense-entries consist of a set of synonyms and a gloss. A gloss consists of a dictionary-style definition and examples demonstrating the use of a synset in a sentence, as shown in Figure 12.1. The figure shows the entries for

12.1.1 Noun

1. **read** (something that is read) *"the article was a very good read"*

12.1.2 Verb

1. **read** (interpret something that is written or printed) *"read the advertisement"; "Have you read Salman Rushdie?"*
2. **read**, say (have or contain a certain wording or form) *"The passage reads as follows"; "What does the law say?"*
3. **read** (look at, interpret, and say out loud something that is written or printed) *"The King will read the proclamation at noon"*
4. **read**, scan (obtain data from magnetic tapes) *"This dictionary can be read by the computer"*
5. **read** (interpret the significance of, as of palms, tea leaves, intestines, the sky; also of human behaviour) *"She read the sky and predicted rain"; "I can't read his strange behavior"; "The fortune teller read his fate in the crystal ball"*
6. take, **read** (interpret something in a certain way; convey a particular meaning or impression) *"I read this address as a satire"; "How should I take this message?"; "You can't take credit for this!"*
7. learn, study, **read**, take (be a student of a certain subject) *"She is reading for the bar exam"*
8. **read**, register, show, record (indicate a certain reading; of gauges and instruments) *"The thermometer showed thirteen degrees below zero"; "The gauge read `empty'"*
9. **read** (audition for a stage role by reading parts of a role) *"He is auditioning for `Julius Caesar' at Stratford this year"*
10. **read** (to hear and understand) *"I read you loud and clear!"*
11. understand, **read**, interpret, translate (make sense of a language) *"She understands French"; "Can you read Greek?"*

Figure 12.1 WordNet 2.0 entry for 'read'

[1] http://wordnet.princeton.edu/

the word 'read'. 'Read' has one sense as a noun and 11 senses as a verb. Glosses help differentiate meanings. Figures 12.2, 12.3, and 12.4 show some of the relationships that hold between nouns, verbs, and adjectives and adverbs. Nouns and verbs are organized into hierarchies based on the hypernymy/hyponymy relation, whereas adjectives are organized into clusters based on antonym pairs (or triplets). Figure 12.5 shows a hypernym chain for 'river' extracted from WordNet. Figure 12.6 shows the troponym relations for the verb 'laugh'.

Relation	*Definition*	*Example*
Hypernym	From concepts to super-ordinates	oak→ tree
Hyponym	From concepts to subtypes	oak→ white oak
Meronym	From wholes to parts	tree → trunk
Holonym	From parts to wholes	trunk → tree
Antonym	Opposites	victory → defeat

Figure 12.2 Noun relations in WordNet

Relation	*Definition*	*Example*
Hypernym	From events to super-ordinate events	wander→ travel
Troponym	From events to their subtypes	walk → stroll
Entails	From events to the events they entail	snore → sleep
Antonym	Opposites	increase → decrease

Figure 12.3 Verb relations in WordNet

Relation	*Definition*	*Example*
Antonym (adjective)	Opposite	heavy → light
Antonym (adverb)	Opposite	quickly → slowly

Figure 12.4 Adjective and adverb relations in WordNet

1 sense of 'river'
Sense 1
river — (a large natural stream of water (larger than a creek); 'the river was navigable for 50 miles')
=> stream, watercourse — (a natural body of running water flowing on or under the earth)
=> body of water, water — (the part of the earth's surface covered with water (such as a river or lake or ocean); 'they invaded our territorial waters'; 'they were sitting by the water's edge')
=> thing — (a separate and self-contained entity)
=> entity — (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

Figure 12.5 Hypernym chain for 'river'

WordNet is freely and publicly available for download from *http://wordnet.princeton.edu/obtain.*

WordNets for other languages have also been developed, e.g., EuroWordNet and Hindi WordNet. EuroWordNet covers European languages, including English, Dutch, Spanish, Italian, German, French, Czech, and Estonian. Other than language internal relations, it also contains multilingual relations from each WordNet to English meanings.

Hindi WordNet has been developed by CFILT (Resource Center for Indian Language Technology Solutions), IIT Bombay.[2] Its database consists of more than 26,208 synsets and 56,928 Hindi words.[3] It is organized using the same principles as English WordNet but includes some Hindi specific relations (e.g., causative relations). A total of 16 relations have been used in Hindi WordNet. Each entry consists of synset, gloss, and position of synset in ontology. Figure 12.7 shows the Hindi WordNet entry for the word 'आकांक्षा' (*aakanksha*).

Sense 1
laugh, express joy, express mirth — (produce laughter)
=> bray — (laugh loudly and harshly)
=> bellylaugh — (laugh a deep, hearty laugh)
=> roar, howl — (laugh unrestrainedly and heartily)
=> snicker, snigger — (laugh quietly)
=> giggle, titter — (laugh nervously; 'The girls giggled when the rock star came into the classroom')
=> break up, crack up — (laugh unrestrainedly)
=> cackle — (emit a loud, unpleasant kind of laughing)
=> guffaw, laugh loudly — (laugh boisterously)
=> chuckle, chortle, laugh softly — (laugh quietly or with restraint)
=> convulse — (be overcome with laughter)
=> cachinnate — (laugh loudly and in an unrestrained way)

Figure 12.6 Troponym relation for the word 'laugh'

Hindi WordNet can be obtained from the URL http://*www.cfilt.iitb.ac.in/wordnet/webhwn/.*

CFLIT has also developed a Marathi WordNet. Figure 12.8 shows the Marathi WordNet (*http://www.cfilt.iitb.ac.in/wordnet/webmwn/wn.php*) entry for the word 'पाव' (*pau*).

12.2.1 Applications of WordNet

WordNet has found numerous applications in problems related with IR and NLP. Some of these are discussed here.

[2]http://www.cfilt.iitb.ac.in/
[3]Hindi WordNet Documentation http://www.cfilt.iitb.ac.in/hindi-wordnet-license01.pdf

Concept Identification in Natural Language

WordNet can be used to identify concepts pertaining to a term, to suit them to the full semantic richness and complexity of a given information need.

Word Sense Disambiguation

WordNet combines features of a number of the other resources commonly used in disambiguation work. It offers sense definitions of words, identifies synstes of synonyms, defines a number of semantic relations and is freely available. This makes it the (currently) best known and most utilized resource for word sense disambiguation. One of the earliest attempts to use WordNet for word sense disambiguation was in IR by Voorheese (1993). She used WordNet noun hierarchy (hypernym / hyponym) to achieve disambiguation. A number of other researchers have also used WordNet for the same purpose (Resnik 1995, 1997; Sussna 1993).

1. (R) अपेक्षा, आकांक्षा, अन्वेक्षा – किसी पर भरोसा रखने की क्रिया कि अमुक कार्य उसके द्वारा हो जायेगा " हर पिता की अपने पुत्र से यह अपेक्षा रहती है कि वह अपने जीवन में सफल हो "
2. (R) इच्छा, अभिलाषा, आकांक्षा, ख्वाहिश, आरज़ू, तमन्ना, कामना, तलब, चेष्टा, हसरत, मुराद, पिपासा, प्यास, तृष्णा, मनोकामना, मनोकामना, मनोवांछा, मनोरथ, मनोभावना, मरज़ी, रजा, मर्ज़ी, मन, रज़ा, मंशा, लिप्सा, लालसा, तृषा, चाह, अरमान, क्षुधा, भूख, भूक, छुधा, हवस, स्पृहा, अभीप्सा, अनु, अपेक्षिता, अभिकांक्षा, अभिकाम, वांछा, वाञ्छा, वाञ्छा, अभिध्या, अभिलाष, अभिप्रीति, अभिमत, अभिमतता, अभिमति, अभिलास, अभिलासा, अभिलाखा, अभिलाख, अभिलाखना — वह मनोवृत्ति जो किसी बात या वस्तु की प्राप्ति की ओर ध्यान ले जाती है " इंसान की हर इच्छा पूरी नही होती / उसकी ज्ञान पिपासा बढ़ती जा रही है / मेरा आज खाने का मन नहीं है "

Figure 12.7 WordNet entry for the Hindi word आकांक्षा (*aakanksha*)

1. (R) पाव एक चतुर्थांश-चौथा भाग "मी बाजारातून एक पाव चुरमुरे आणले"
2. (R) पाव, पावरोटी-गव्हाचे पीठ आंबवून केलेल एक खाद्यविशेष "मुंबईत बरेच लोक पाव खाऊन गुजराण करतात"
3. (R) पाव-पावाचे वजन "दुकानदार चहाच्या पूडीचे वजन करण्यासाठी पाव शोधत आहे"
4. (R) पाव-एक चतुर्थांश बाटली दारू "एक पाव प्यायल्यावर तो वटवट करू लागलाए"

Figure 12.8 WordNet entry for the Marathi word पाव (*pau*)

Automatic Query Expansion

WordNet semantic relations can be used to expand queries so that the search for a document is not confined to the pattern-matching of query terms, but also covers synonyms. The work performed by Voorhees (1994) is based on the use of WordNet relations, such as synonyms, hypernyms, and hyponyms, to expand queries.

4) ***Document Structuring and Categorization***

The semantic information extracted from WordNet, and WordNet conceptual representation of knowledge, have been used for text categorization (Scott and Matwin 1998).

5) ***Document Summarization***

WordNet has found useful application in text summarization. The approach presented by Barzilay and Elhadad (1997) utilizes information from WordNet to compute lexical chains.

12.3 FRAMENET

FrameNet[4] is a large database of semantically annotated English sentences. It is based on principles of frame semantics. It defines a tagset of semantic roles called the frame element. Sentences from the British National Corpus are tagged with these frame elements. The basic philosophy involved is that each word evokes a particular situation with particular participants. FrameNet aims at capturing these situations through case-frame representation of words (verbs, adjectives, and nouns). The word that invokes a frame is called *target word* or *predicate*, and the participant entities are defined using semantic roles, which are called *frame elements*. The FrameNet ontology can be viewed as a semantic level representation of predicate argument structure.

Each frame contains a main lexical item as predicate and associated frame-specific semantic roles, such as AUTHORITIES, TIME, and SUSPECT in the ARREST frame, called frame elements. As an example, consider sentence (12.1) annotated with the semantic roles AUTHORITIES and SUSPECT. The target word in sentence (12.1) is 'nab' which is a verb in the ARREST frame.

[Authorities The police] nabbed [suspect the snatcher]. (12.1)

A COMMUNICATON frame has the semantic roles ADDRESSEE, COMMUNICATOR, TOPIC, and MEDIUM. Figure 12.9 shows the core and non-core frame elements of the COMMUNICATION frame, along with other details. A JUDGEMENT frame contains roles such as JUDGE, EVALUEE, and REASON. A frame may inherit roles from another frame. For example, a STATEMENT frame may inherit from a COMMUNICATION frame; it contains roles such as SPEAKER, ADDRESSEE, and MESSAGE. The following sentences show some of these roles:

[Judge She] [Evaluee blames the police] [Reason for failing to provide enough protection]. (12.2)

[Speaker She] told [Addressee me] [Message 'I'll return by 7:00 pm today']. (12.3)

[4] http://framenet.icsi.berkeley.edu/

12.3.1 FrameNet Applications

Gildea and Jurafsky (2002) and Kwon et al. (2004) used FrameNet data for automatic semantic parsing. The shallow semantic role obtained from FrameNet can play an important role in information extraction. For example, a semantic role makes it possible to identify that the theme role played by 'match' is same in sentences (12.4) and (12.5) though the syntactic role is different.

The umpire stopped the match. (12.4)

The match stopped due to bad weather. (12.5)

In sentence (12.4), the word 'match' is the object, while it is the subject in sentence (12.5).

Semantic roles may help in the question-answering system. For example, the verb 'send' and 'receive' would share the semantic roles SENDER, RECIPIENT, GOODS, etc., (Gildea and Jurafsky 2002) when defined with respect to a common TRANSFER frame. Such common frames allow a question-answering system to answer a question such as 'Who sent packet to Khushbu?' using sentence (12.6).

Khushbu received a packet from the examination cell. (12.6)

Other applications include IR (Mohit and Narayanan 2003), interlingua for machine translation, text summarization, and word sense disambiguation.

Communication

Frame Elements

Core:

Addressee [Add]	Receiver of Message from the Communicator.
Communicator [Com]	The person conveying (written or spoken) a message to another person.
Message [Msg]	A proposition or set of propositions that the Communicator wants the Addressee to convey
Topic [Top]	The entity that the proposition(s) are about.

Non-core:

Amount_of_information [Amo]	The amount of information exchanged when communication occurs.
Depictive [Dep-Act]	The Depictive describes the state of the Communicator.
Duration []	The length of time during which the communication takes place.
Manner [Manr]	The Manner in which the Communicator communicates.
Means [Mns]	The Means by which the Communicator communicates.
Medium [Medium]	The physical or abstract setting in which the Message is conveyed.
Time []	The time at which the communication takes place.

Inherits From:

Is Inherited By: Communication_noise, Statement

Subframe of:

Has Subframes:

Uses: Topic

Is Used By: Claim_ownership, Communication_response, Contacting, Deny_permission, Discussion, Hear, Questioning, Reasoning, Reporting, Request, etc

Is Inchoative of:

Is Causative of:

See Also:

Sample Predi

12.4 STEMMERS

As discussed in Chapter 3, stemming, often called conflation, is the process of reducing inflected (or sometimes derived) words to their base or root form. The stem need not be identical to the morphological base of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Stemming is useful in search engines for query expansion or indexing and other NLP problems. Stemming programs are commonly referred to as stemmers. The most common algorithm for stemming English is Porter's algorithm[5] (Porter 1980). Other existing stemmers include Lovins[6] stemmer (Lovins 1968) and a more recent one called the Paice/Husk stemmer[7] (Paice 1990). Figure 12.10 shows a sample text and output produced using these stemmers.

Input Text:
Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation.

Output:
Lovins stemmer: such an analys can rev feature that ar not eas vis from the vari in th individu gen and can lead to a picture of expres that is mor biolog transpar and acces to interpres
Porter's stemmer: such an analysi can reveal feature that ar not easily visible from the variat in the individu gene and can lead to a picture of express that is more biolog transpar and access to interpret
Paice stemmer: Such an analys can rev feat that are not easy vis from the vary in the individ gen and can lead to a pict of express that is mor biolog transp and access to interpret

Figure 12.10 Stemmed text using different stemmers

12.4.1 Stemmers for European Languages

There are many stemmers available for English and other languages. Snowball[8] presents stemmers for English, Russian, and a number of other European languages, including French, Spanish, Portuguese, Hungarian, Italian, German, Dutch, Swedish, Norwegian, Danish, and Finnish. The links for stemming algorithms for these languages can be found at *http://snowball.tartarus.org/texts/stemmersoverview.html.*

[5]http://tartarus.org/~martin/PorterStemmer/
[6]http://sourceforge.net/project/showfiles.php?group_id=24260
[7]http://www.comp.lancs.ac.uk/computing/research/stemming/Links/implementations.htm
[8]http://snowball.tartarus.org/

12.4.2 Stemmers for Indian Languages

Standard stemmers are not yet available for Hindi and other Indian languages. The major research on Hindi stemming has been accomplished by Ramanathan and Rao (2003) and Majumder et al. (2007). Ramanathan and Rao (2003) based their work on the use of handcrafted suffix lists. Majumder et al. (2007) used a cluster-based approach to find classes of root words and their morphological variants. They used a task-based evaluation of their approach and reported that stemming improves recall for Indian languages. Their observation on Indian languages was based on a Bengali data set. The Resource Centre of Indian Language Technology (CFILT), IIT Bombay has also developed stemmers for Indian languages, which are available at *http://www.cfilt.iitb.ac.in.*

12.4.3 Stemming Applications

Stemmers are common elements in search and retrieval systems such as Web search engines. Stemming reduces the variants of a word to same stem. This reduces the size of the index and also helps retrieve documents that contain variants of a query terms. For example, a user issuing a query for documents on 'astronauts' would like documents on 'astronaut' as well. Stemming permits this by reducing both versions of the word to the same stem. However, the effectiveness of stemming for English query systems is not too great, and in some cases may even reduce precision.

Text summarization and text categorization also involve term frequency analysis to find features. In this analysis, stemming is used to transform various morphological forms of words into their stems.

12.5 PART-OF-SPEECH TAGGER

Part-of-speech tagging is used at an early stage of text processing in many NLP applications such as speech synthesis, machine translation, IR, and information extraction. In IR, part-of-speech tagging can be used in indexing (for identifying useful tokens like nouns), extracting phrases and for disambiguating word senses. The rest of this section presents a number of part-of-speech taggers that are already in place.

12.5.1 Stanford Log-linear Part-of-Speech (POS) Tagger

This POS Tagger is based on maximum entropy Markov models. The key features of the tagger are as follows:

(i) It makes explicit use of both the preceding and following tag contexts via a dependency network representation.

(ii) It uses a broad range of lexical features.
(iii) It utilizes priors in conditional log-linear models.

The reported accuracy of this tagger on the Penn Treebank WSJ is 97.24%, which amounts to an error reduction of 4.4% on the best previous single automatically learned tagging result (Tuotanova et al. 2003). Details on the tagger can be found at the link *http://nlp.stanford.edu/software/tagger.shtml*.

12.5.2 A Part-of-Speech Tagger for English[9]

This tagger uses a bi-directional inference algorithm for part-of-speech tagging. It is based on maximum entropy Markov models (MEMM). The algorithm can enumerate all possible decomposition structures and find the highest probability sequence together with the corresponding decomposition structure in polynomial time. Experimental results of this part-of-speech tagger show that the proposed bi-directional inference methods consistently outperform unidirectional inference methods and bi-directional MEMMs give comparable performance to that achieved by state-of-the-art learning algorithms, including kernel support vector machines (Tsuruoka and Tsujii 2005).

12.5.3 TnT tagger[10]

Trigrams'n'Tags or TnT (Brants 2000) is an efficient statistical part-of-speech tagger. This tagger is based on hidden Markov models (HMM) and uses some optimization techniques for smoothing and handling unknown words. It performs at least as well as other current approaches, including the maximum entropy framework. Table 12.1 shows tagged text of document #93 of the CACM collection.

Table 12.1 Doc #93 of CACM collection tagged using TnT tagger

A	DT	simple	JJ
technique	NN	algebraic	JJ
is	VBZ	formulas	NNS
shown	VBN	into	IN
for	IN	a	DT
enabling	VBG	three	CD
a	DT	address	NN
computer	NN	computer	NN
to	TO	code	NN
translate	VB		

[9]http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/postagger/
[10]http://www.coli.uni-saarland.de/~thorsten/tnt/

12.5.4 Brill Tagger

Brill (1992) described a trainable rule-based tagger that obtained performance comparable to that of stochastic taggers. It uses transformation-based learning to automatically induce rules. A number of extensions to this rule-based tagger have been proposed by Brill (1994). He describes a method for expressing lexical relations in tagging that stochastic taggers are currently unable to express. It implements a rule-based approach to tagging unknown words. It demonstrates how the tagger can be extended into a k-best tagger, where multiple tags can be assigned to words in some cases of uncertainty. Brill tagger is available for download at the link *http://www.cs.jhu.edu/~brill/RBT1_14.tar.Z*.

12.5.5 CLAWS Part-of-Speech Tagger for English

Constituent likelihood automatic word-tagging system (CLAWS) is one of the earliest probabilistic taggers for English. It was developed at the University of Lancaster (*http://ucrel.lancs.ac.uk/claws*). The latest version of the tagger, CLAWS4, can be considered a hybrid tagger as it involves both probabilistic and rule-based elements. It has been designed so that it can be easily adapted to different types of text in different input formats. CLAWS has achieved 96–97% accuracy. The precise degree of accuracy varies according to the type of text. For more information on the CLAWS tagger, see Garside (1987), Leech, Garside, and Bryant (1994), Garside (1996), and Garside and Smith (1997).

12.5.6 Tree-Tagger

Tree-Tagger (Schmidt 1994) is a probabilistic tagging method. It avoids problems faced by the Markov model methods when estimating transition probabilities from sparse data, by using a decision tree to estimate transition probabilities. The decision tree automatically determines the appropriate size of the context to be used in estimation. The reported accuracy for the tagger is above 96% on the Penn-Treebank WSJ corpus. The tagger is available at the link *http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html*.

12.5.7 ACOPOST: A Collection of POS Taggers[11]

ACOPOST is a set of freely available POS taggers. The taggers in the set are based on different frameworks. The programs are written in C. ACOPOST currently consists of the following four taggers.

[11] http://acopost.sourceforge.net/

Maximum Entropy Tagger (MET)

This tagger is based on a framework suggested by Ratnaparkhi (1997). It uses an iterative procedure to successively improve parameters for a set of features that help to distinguish between relevant contexts.

Trigram Tagger (T3)

This tagger is based on HMM. The states in the model are tag pairs that emit words. The technique has been suggested by Rabiner (1990) and the implementation is influenced by Brants (2000).

Error-driven Transformation-based Tagger (TBT)

This tagger is based on the transformation-based tagging approach proposed by Brill (1993). It uses annotated corpuses to learn transformation rules, which are then used to change the assigned tag using contextual information.

Example-based Tagger (ET)

The underlying assumption of example-based models (also called memory-based, instance-based or distance-based models) is that cognitive behaviour can be achieved by looking at past experiences that match the current problem, instead of learning and applying abstract rules. This framework has been suggested for NLP by Daelemans et al. (1996).

12.5.7 POS Tagger for Indian Languages

The automatic text processing of Hindi and other Indian languages is constrained heavily due to lack of basic tools and large annotated corpuses. Research groups are now focusing on removing these bottlenecks. The work on the development of tools, techniques, and corpora is going on at several places such as CDAC, IIT Bombay, IIIT Hyderabad, University of Hyderabad, CIIL Mysore, and University of Lancaster. IIT Bombay is involved in the development of morphology analysers and part-of-speech taggers for Hindi and Marathi. Both these languages have rich morphological structures. Their approach is based on *bootstrapping on a small corpus tagged* by a rule-based tagger and then applying statistical techniques to train a machine. More information can be found at *http://ltrc.iiit.net* and *www.cse.iitb.ac.in*. Work on Urdu part-of-speech taggers has been reported by Hardie (2003) and Baker et al. (2004).

Research corpora have been developed for a number of NLP-related tasks. In the following section, we point out few of the available standard document collections for a variety of NLP-related tasks, along with their Internet links.

12.6.1 IR Test Collection

We have already provided a list of IR test document collection in Chapter 9. Glasgow University, UK, maintains a list of freely available IR test collections. Table 12.2 lists the sources of those and few more IR test collections.

LETOR (learning to rank) is a package of benchmark data sets released by Microsoft Research Asia. It consists of two datasets OHSUMED and TREC (TD2003 and TD2004). LETOR is packaged with extracted features for each query-document pair in the collection, baseline results of several state-of-the-art learning-to-rank algorithms on the data and evaluation tools. The data set is aimed at supporting future research in the area of learning ranking function for information retrieval.

Table 12.2 IR test collection

LETOR	*http://research.microsoft.com/users/tyliu/LETOR/*
LISA	
CACM	
CISI	
MEDLINE	*http://www.dcs.gla.ac.uk/idom/ir_resources/test_collections/*
Cranfield	
TIME	
ADI	

12.6.2 Summarization Data

Evaluating a text summarizing system requires existence of 'gold summaries'. DUC provides document collections with known extracts and abstracts, which are used for evaluating performance of summarization systems submitted at TREC conferences. Figure12.11 shows a sample document and its extract from DUC 2002 summarization data.

<DOC>
<DOCNO> AP880911-0016 </DOCNO>
<FILEID>AP-NR-09-11-88 0423EDT</FILEID>
<FIRST>r i BC-HurricaneGilbert 09-11 0339</FIRST>
<SECOND>BC-Hurricane Gilbert,0348</SECOND>
<HEAD>Hurricane Gilbert Heads Toward Dominican Coast</HEAD>
<BYLINE>By RUDDY GONZALEZ</BYLINE>
<BYLINE>Associated Press Writer</BYLINE>
<DATELINE>SANTO DOMINGO, Dominican Republic (AP) </DATELINE>
<TEXT>

Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains, and high seas.

The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.

"There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Saturday.

Cabral said residents of the province of Barahona should closely follow Gilbert's movement. An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo.

Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.

The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm.

The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.

Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet to Puerto Rico's south coast. There were no reports of casualties.

San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.

On Saturday, Hurricane Florence was downgraded to a tropical storm and its remnants pushed inland from the U.S. Gulf Coast. Residents returned home, happy to find little damage from 80 mph winds and sheets of rain.

Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane. The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.
</TEXT>
</DOC>

Extract

Tropical Storm Gilbert in the eastern Caribbean strengthened into a hurricane Saturday night. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday to be about 140 miles south of Puerto Rico and 200 miles southeast of Santo Domingo. It is moving westward at 15mph with a broad area of cloudiness and heavy weather with sustained winds of 75mph gusting to 92mph. The Dominican Republic's Civil Defense alerted that country's heavily populated south coast and the National Weather Service in San Juan, Puerto Rico issued a flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.

Figure 12.11 Sample document from DUC 2002 and its extract

12.6.3 Word Sense Disambiguation

SEMCOR[12] is a sense-tagged corpus used in disambiguation. It is a subset of the Brown corpus, sense-tagged with WordNet synsets. Open Mind Word Expert[13] attempts to create a very large sense-tagged corpus. It collects word sense tagging from the general public over the Web.

12.6.4 Asian Language Corpora

The multilingual EMILLE corpus is the result of the enabling minority language engineering (EMILLE) project at Lancaster University, UK. The project focuses on generation of data, software resources and basic language engineering tools for the NLP of south Asian languages. Central Institute for Indian Languages (CIIL), the Indian partner in the project, extended the set of target languages to include a number of Indian languages. CIIL provides a wider range of data in these languages from a wide range of genres. The data sources that EMILLE made available include monolingual written and spoken corpuses, parallel and annotated corpuses. The full EMILLE/CIIL corpus is available for free, but for research use only, at the link *http://www.elda.org/catalogue/en/text/W0037.html.* Further details about the corpus can be found in the manual at the site *http://www.emille.lancs.ac.uk/manual.pdf.*

Corpus building in these languages is constrained by the scarcity of repositories of electronic text. The monolingual corpus includes written data for 14 South Asian languages and spoken data for five languages (Hindi, Bengali, Gujrati, Punjabi, and Urdu). The spoken corpus was constructed from radio broadcasts on the BBC Asia network. The parallel corpus contains English text and its translation in five languages. The text includes UK government advice leaflets which are published in multiple languages. The corpus is aligned at sentence level. The parallel corpus provided by EMILLE corpus is a valuable resource for statistical machine translation research. The annotated component includes Urdu data annotated for part-of-speech tagging, and a Hindi corpus annotated to show nature of demonstrative use.

12.7 JOURNALS AND CONFERENCES IN THE AREA

A wide number of conference proceedings and journals report research in the various areas of NLP. Most notable among them are those associated with Association for Computing Machinery (ACM), Association for

[12] http://www.cs.unt.edu/~rada/downloads.html#semcor
[13] http://teach-computers.org.

Computational Linguistics (ACL), its European counterpart EACL, Recherche d'Information Assistie par Ordinateur (RIAO) and the International Conferences on Computational Linguistics (COLING). The ACM SIGIR Conference is one of the major conferences held on research and development in information retrieval. It provides the international forum for dissemination of research and demonstration of new systems and techniques. The 30th Annual International ACM SIGIR Conference[14] was held on 23–27 July 2007 at Amsterdam. The Proceedings of Text Retrieval Conferences (TRECs)[15] are another important source of information. These proceedings report results from standardized evaluations organized by the US government. The TRECs have been organized regularly since 1992 as a part of the TIPSTER text retrieval. They were earlier known as the Document Understanding Conference or Message Understanding Conferences. The conference series is sponsored by the National Institute of Standards and Technology (NIST) with additional support from other US government agencies. The ACM Special Interest Group on Information Retrieval (ACM-SIGIR) focuses on IR related tasks, and ECIR is its European counterpart. The NTCIR (NII test collection for IR) focuses on information retrieval with Japanese and other Asian languages.

KES[16] International Conferences in Knowledge-Based and Intelligent Engineering & Information Systems have been a regular feature since 1997. The conference mainly focuses on applications of intelligent systems. The topics covered by KES includes general intelligent topics like neural networks, fuzzy techniques, genetic algorithms, knowledge representation and management, applications using intelligent techniques (e.g., speech processing and synthesis and NLP) and emerging intelligent technologies like intelligent information retrieval, intelligent web mining and applications, intelligent user interfaces, etc.

HLT-NACCL is sponsored by the North American chapter of the Association for Computational Linguistics.

The *Journal of Computational Linguistics* is a leading premier publication focussing on theoretical and linguistics aspects. More practical applications are covered in the *Natural Language Engineering Journal. Information Retrieval* by Kluwer, *Information Processing and Management* by Elsevier, ACM's *Transactions on Information Systems* (TOIS), *Journal of American Society for Information Sciences* are major journals covering a wide range of information

[14] http://www.sigir2007.org/
[15] http://trec.nist.gov/
[16] http://www.kesinternational.org/conferences.php

Research, International Journal of Information Technology and Decision Making (World Scientific), and *Journal of Digital Information Management and Information System.*

A few AI publications also report work on language processing. Among these are *Artificial Intelligence, Computational Intelligence,* IEEE's *Transaction on Intelligent Systems,* and *Journal of AI Research.*

SUMMARY

- Lexical resources such as WordNet and FrameNet can be used in a number of NLP-related tasks.
- Stemmers are useful in a number of information processing tasks such as information retrieval, text summarization, and text categorization.
- Widely known stemmers include Porter's and Lovins stemmers.
- Part-of-speech tagger is used to assign a part-of-speech, such as noun, verb, pronoun, preposition, adverb, and adjective, to each word in a sentence (or text).
- Taggers include stanford log-linear part-of-speech tagger, TnT, CLAWS, and Brill's tagger.
- TREC and SIGIR conferences offer useful resources for a number of information processing-related tasks.

REFERENCES

Ananthkrishnan, R. and Durgesh Rao, 2003, 'A lightweight stemmer for Hindi,' Workshop on Computational Linguistics for South Asian Languages, *The 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, ACL, Morristown, NJ.

Baker, P., A. Hardie, A.M. McEnery, and B.D. Jayaram, 2004, 'Corpus linguistics and South Asian languages: corpus creation and tool development,' *Literary and Linguistic Computing*, 19(4), 509–24.

Barzilay, Regina and Michael Elhadad, 1997, 'Using lexical chains for text summarization,' *Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97)*, ACL, Madrid.

Brants, Thosrten, 2000, 'TnT–as statistical part-of-speech tagger,' *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000)*, Seattle, WA.

Brill E., 1992, 'A simple rule-based part-of-speech tagger,' *Proceedings of the Third Conference on Applied Natural Language Processing*, ACL, Budapest, Hungary.

APPENDIX A

PENN TREEBANK TAGSET

CC	Coordinating conjunction—for example and, but, and or
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal—for example can, could, might, and may
NN	Noun, singular or mass
NNP	Proper noun, singular
NNPS	Proper noun, plural
NNS	Noun, plural
PDT	Pre-determiner—for example all and both when they precede an article
POS	Possessive ending—for example nouns ending in 's'
PRP	Personal pronoun—for example I, me, you, and he
PRP$	Possessive pronoun—for example my, your, mine, and yours
RB	Adverb—most words that end in -ly as well as degree words such as quite, too, and very
RBR	Adverb, comparative—adverbs with the comparative ending -er, with a strictly comparative meaning

RBS	Adverb, superlative
RP	Particle
SYM	Symbol—should be used for mathematical, scientific, or technical symbols
TO	to
UH	Interjection—for example uh, well, yes, and my
VB	Verb, base form—subsumes imperatives, infinitives, and subjunctives
VBD	Verb, past tense—includes the conditional form of the verb to be
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-third person singular present
VBZ	Verb, third person singular present
WDT	Wh-determiner—for example which and that when it is used as a relative pronoun
WP	Wh-pronoun—for example what, who, and whom
WP$	Possessive wh-pronoun
WRB	Wh-adverb—for example how, where, and why

Punctuation tags
#
$
“
(
)
,
.
:
“ ”

APPENDIX B

PORTER STEMMER

The Porter stemming algorithm consists of condition/action pairs. Actions are in the form of rewrite rules. The conditions may be on stems, suffix, or rules. The stem conditions take either of the following form:

(i) $m = 0$, 1, or 2.

(ii) Stem contains or ends with (pattern).

where m, the measure, is the number of vowel–consonant (VC) sequence. For example, for the word in 'tea' and 'coffee', $m = 0$ as the number of vowel–consonant sequence is 0, whereas for the word 'astronaut', $m = 3$ ('as', 'on', and 'ut').

The patterns are of the form:

$*<x>$	stem ends with a given letter x
$*v*$	stem contains a vowel
$*d$	stem ends in a double consonant
$*o$	stem ends in a consonant–vowel–consonant sequence, where the final consonant is not w, x, or y.

Suffix condition takes the form 'current_suffix = = pattern' and rule conditions take the form 'rule was used'. Action rules are of the form 'old_suffix → new_suffix'.

The steps involved in Porter stemming algorithm are given below:

Step 1: Deal with plurals and past participles.

Step 1a:

Action rules	*Examples*
SSES → SS	dresses → dress
IES → I	patties → patti
SS → SS	success → success
S → null	girls → girl

Step 1b:

Conditions	*Action rules*	*Examples*
$(m > 0)$	EED → EE	succeed → succee
(*v*)	ED → null	deed → deed walked → walk
(*v*)	ING → null	fled → fled playing → play sing → sing

Step 1b1:

Conditions	*Action rules*	*Examples*
Null	AT → ATE	rotat(ed) → rotate
Null	BL → BLE	struggl(ed) → struggle
Null	IZ → IZE	recogniz(ed) → recognize
(*d and not	double letter	letter scann(ed) → scan
(*L or *S or *Z))	→ single	call(ing) → call
		miss(ing) → miss
$(m = 1$ and *o)	null → E	trail(ing) → trail
		pil(ing) → pile

Step 1c:

(*v*) Y → I	lazy → lazi
	spy → spy

Step 2:

$(m > 0)$ ATIONAL → ATE	rotational → rotate
$(m > 0)$ TIONAL → TION	proportional → proportion
$(m > 0)$ ENCI → ENCE	valenci → valence
$(m > 0)$ ANCI → ANCE	hesitanci → hesitance
$(m > 0)$ IZER → IZE	recognizer → recognize
$(m > 0)$ ABLI → ABLE	stabli → stable
$(m > 0)$ ALLI → AL	practicalli → practical
$(m > 0)$ ENTLI → ENT	differentli → different
$(m > 0)$ ELI → E	vileli → vile
$(m > 0)$ OUSLI → OUS	previousli → previous
$(m > 0)$ IZATION → IZE	privatization → privatize
$(m > 0)$ ATION → ATE	predication → predicate
$(m > 0)$ ATOR → ATE	dictator → dictate
$(m > 0)$ ALISM → AL	socialism → social

$(m > 0)$ IVENESS → IVE	comprehensiveness → comprehensive
$(m > 0)$ FULNESS → FUL	successfulness → successful
$(m > 0)$ OUSNESS → OUS	obviousness → obvious
$(m > 0)$ ALITI → AL	realiti → real
$(m > 0)$ IVITI → IVE	transitiviti → transitive
$(m > 0)$ BILITI → BLE	reliabiliti → reliable

Step 3:

$(m > 0)$ ICATE → IC	replicate → replic
$(m > 0)$ ATIVE → null	formative → form
$(m > 0)$ ALIZE → AL	conceptualize → conceptual
$(m > 0)$ ICITI → IC	electriciti → electric
$(m > 0)$ ICAL → IC	aeronautical → aeronautic
$(m > 0)$ FUL →	hopeful → hope
$(m > 0)$ NESS →	goodness → good

Step 4:

$(m > 1)$ AL →	revival → reviv
$(m > 1)$ ANCE →	allowance → allow
$(m > 1)$ ENCE →	preference → prefer
$(m > 1)$ ER →	hardliner → hardlin
$(m > 1)$ IC →	endoscopic → endoscop
$(m > 1)$ ABLE →	adjustable → adjust
$(m > 1)$ IBLE →	defensible → defens
$(m > 1)$ ANT →	irritant → irrit
$(m > 1)$ EMENT →	replacement → replac
$(m > 1)$ MENT →	adjustment → adjust
$(m > 1)$ ENT →	dependent → depend
$(m > 1$ and (*S or *T)) ION →	adoption → adopt
$(m > 1)$ OU →	homologou → homolog
$(m > 1)$ ISM →	communism → commun
$(m > 1)$ ATE →	activate → activ
$(m > 1)$ ITI →	singulariti → singular
$(m > 1)$ OUS →	anologous → homolog
$(m > 1)$ IVE →	defective → defect
$(m > 1)$ IZE →	equalize → equal

The suffixes are now removed. All that remains is a little tidying up.

Step 5a:

$(m > 1)$ E →	equate → equat date → date
$(m = 1$ and not *o) E → null	cease → ceas

Step 5b:

$(m > 1$ and *d and *L)	null → single letter controll → control roll → roll