

4 Pattern Recognition

4.1 Introduction

As it was learnt from the foregoing chapters, information extraction concerns the detection and recognition of certain information, and it relies on pattern recognition methods. Pattern recognition (also known as classification or pattern classification) aims at classifying data (patterns) based on either a priori knowledge that is acquired by human experts or on knowledge automatically learned from data. A system that automatically sorts patterns into classes or categories is called a *pattern classifier*. The classification patterns are recognized as a combination of features and their values. In case of information extraction the features are textual characteristics that can be identified or measured, and that are assumed to have a discriminative value when sorting patterns into semantic classes.

As seen in the previous chapter, in its early days information extraction from texts relied on symbolic, handcrafted knowledge. Information was extracted using a set of patterns in the form of rules or a grammar, and a recognizer called an automaton parsed the texts with the objective to find constructions that conform with the grammar and that were translated into semantic concepts or relations. More recent technologies often use feature vectors as the input of *statistical* and *machine learning algorithms* in order to detect the classification patterns. Supervised, unsupervised and weakly supervised learning algorithms are common. The machine learning algorithms relieve the burden of the manual knowledge acquisition. The algorithms exhibit an additional advantage. Instead of a deterministic translation of text units into semantic classes as seen in the previous chapter, the approaches usually allow a *probabilistic class assignment*, which is useful if we want to make probabilistic inferences based on the extracted information. For instance, information retrieval models use probabilistic models such as Bayesian networks and reasoning with uncertainty when inferring the relevance of a document to a query. After all, when we humans read and understand a text, we make many (sometimes uncertain)

inferences with the content of a text in combination with additional world knowledge, the background knowledge of the reader, and his or her information goals (Graesser and Clark, 1985). Any intelligent information processing system that relies on extracted information should incorporate uncertainties about the information extracted.

Before proceeding to the next chapters that discuss prevalent pattern recognition methods used in information extraction, several important questions have to be answered. *What are the information units and their relations that we want to detect in the texts and classify? How do we conveniently detect these information units? What are the classification schemes used in information extraction? How can an information unit be described with a feature vector or other object that captures the necessary feature values for correct classification? How can these features and their values be identified in the texts?*

The aim of the book is to focus on generic and flexible approaches to information extraction. When we answer the above questions, the focus is on technologies that can be used in open domain settings. It will be shown that many of the information extraction tasks require similar types of features and classification algorithms. By stressing what binds the approaches, we hope to promote the development of generic information extraction technology that can be used in many extraction settings. The text of this chapter will be illustrated with many different examples of common extraction tasks such as named entity recognition, coreference resolution, semantic role recognition, relation recognition and timex recognition.

4.2 What is Pattern Recognition?

Pattern recognition classifies objects into a number of classes or categories based on the patterns that objects exhibit (Theodoridis and Koutroumbas 2003). The objects are described with a number of selected features and their values. An object x thus can be described as a *vector of features*:

$$\mathbf{x} = [x_1, x_2, \dots, x_p]^T \quad (4.1)$$

where p = the number of features measured.

The features or attributes together span a multi-variate space called the measurement space or feature space. Throughout the following chapters, features and feature vectors will be treated as random variables and vectors respectively. The measurements exhibit a random variation. This is partly due to the measurement noise of measuring devices and partly to the distinct

characteristics of each feature. When features and their values are identified in natural language text, we might not capture the values correctly because our tools cannot yet cope with all variations and ambiguities a natural language exhibits.

Vectors are not the sole representation format that we use for representing the textual objects. We can also use *structured objects* as representations such as presentations in first-order predicate logic and graphs. A text is often well suited to be represented as a *tree* (e.g., based on its parse or discourse tree), where the relations between features are figured as edges between the nodes, and nodes can contain the attributes of the features.

The classification task can be seen as a two (binary) or multi-class problem. In a *two-class problem*, an object is classified as belonging or not belonging to a particular class and one trains a binary classifier for each class. In a *multi-class problem* the classification task is defined as one multi-class learning problem. It is convenient to learn multiple binary classifiers when the classes are not mutually exclusive. In the information extraction tasks, which we will further consider, classes are often mutually exclusive allowing treating information extraction as a multi-class learning problem.

Pattern recognition methods regard machine learning. The learning algorithm takes the training data as input and selects a hypothesis from the hypothesis space that fits the data. There are many different learning algorithms. The availability or non-availability of training examples determines whether the machine learning is considered as respectively supervised or unsupervised.

In *supervised pattern recognition*, usually a rather large set of classified examples can be used for training the classifier. The feature vectors whose true classes are known and which are used for building the classifier are considered as training examples and form the *training set*. Because in information extraction we work with textual material, the assignment of the true class is usually done by annotating the text with class labels. For instance, in a named entity recognition task proper names can be annotated with entity class labels (see Fig. 4.1).

In supervised pattern recognition the aim is to detect general, but high-accuracy classification patterns in the training set, that are highly predictable to correctly classify new, previously unseen instances of a *test set*. It is important to choose the appropriate training algorithm (e.g., support vector machines, maximum entropy modeling, induction of rules and trees) in compliance with a number of a priori defined constraints on the data (e.g., dependency of features, occurrence of noisy features, size of the feature set, size of the training set, etc...).

```

<HL> <ENAMEX TYPE="ORGANIZATION">Eastern Air</ENAMEX> Proposes
Date For Talks on Pay-Cut Plan</HL>
<DD> <TIMEX TYPE="DATE">01/23/87</TIMEX></DD>
<SO> WALL STREET JOURNAL (J)</SO>
<IN> LABOR TEX AIRLINES (AIR) </IN>
<DATELINE> <ENAMEX TYPE="LOCATION">MIAMI</ENAMEX> </DATELINE>
<TXT>
<p>
<s> <ENAMEX TYPE="ORGANIZATION">Eastern Airlines</ENAMEX> execu-
tives notified union leaders that the carrier wishes to discuss
selective wage reductions on <TIMEX TYPE="DATE">Feb. 3</TIMEX>.
</s>
</p>
<p>
<s> Union representatives who could be reached said they hadn't
decided whether they would respond. </s>
</p>
<p>
<s> By proposing a meeting date, <ENAMEX
TYPE="ORGANIZATION">Eastern</ENAMEX> moved one step closer to-
ward reopening current high-cost contract agreements with its
unions. </s>
<s> The proposal to meet followed an announcement <TIMEX
TYPE="DATE">Wednesday</TIMEX> in which <ENAMEX
TYPE="PERSON">Philip Bakes</ENAMEX>, <ENAMEX
TYPE="ORGANIZATION">Eastern</ENAMEX>'s president, laid out pro-
posals to cut wages selectively an average of <NUMEX
TYPE="PERCENT">29%</NUMEX>. </s>
<s> The airline's three major labor unions, whose contracts
don't expire until year's end at the earliest, have vowed to re-
sist the cuts. </s>
</p>
<p>
<s> Nevertheless, one union official said he was intrigued by
the brief and polite letter, which was hand-delivered by corpo-
rate security officers to the unions. </s>
<s> According to <ENAMEX TYPE="PERSON">Robert Callahan</ENAMEX>,
president of <ENAMEX TYPE="ORGANIZATION">Eastern</ENAMEX>'s
flight attendants union, the past practice of <ENAMEX
TYPE="ORGANIZATION">Eastern</ENAMEX>'s parent, <ENAMEX
TYPE="LOCATION">Houston</ENAMEX>-based <ENAMEX
TYPE="ORGANIZATION">Texas Air Corp.</ENAMEX>, has involved con-
frontation and ultimatums to unions either to accept the car-
rier's terms or to suffer the consequences - in this case,
perhaps, layoffs. </s>
</p>
<p>
<s> "Yesterday's performance was a departure," Mr. <ENAMEX
TYPE="PERSON">Callahan</ENAMEX> said, citing the invitation to
conduct broad negotiations - and the lack of a deadline imposed
by management. </s>
<s> "Frankly, it's a little mystifying." </s>
</p>
</TXT>

```

Fig. 4.1. Annotated sentences from MUC-6 Document No. 870123-0009.

Unsupervised pattern recognition tries to unravel similarities or differences between objects and to group or cluster similar objects. Cluster algorithms are often used for this purpose. Unsupervised learning is a necessity when the classes are not a priori known, when annotated examples are not available or too expensive to produce, or when objects and their features or feature values change very dynamically. For instance, non-pronominal noun phrase coreference resolution across documents in document collections that dynamically change (such as news stories) is an example of where unsupervised learning is useful, because the context features of all noun phrases are very likely to exhibit a large variation over time.

In unsupervised pattern recognition an important focus is on the selection of features. One often relies on knowledge or an appreciation of features that are a priori assumed not to be relevant for the classes sought. In addition, the choice of a suitable function that computes the similarity or distance between two feature vectors is very important as these functions give different results depending on where the feature vectors are located in the feature space (cf. Jones and Furnas, 1987). The choice of a convenient cluster algorithm that clusters the objects into groups is important as well. Here too, the choice is defined by a number of a priori defined constraints on the data, such as the number of feature vectors and their location in the geometrical feature space.

Because of the large variety of natural language expressions it is not always possible to capture this variety by sufficient annotated examples. On the other hand, we have huge amounts of unlabeled data sets in large text collections. Hence, the interest in unsupervised approaches for the semantic classification or in *unsupervised aids* that complement the lack of sufficient training examples.

In the framework of generic technologies for information extraction, it is important that the classification or extraction patterns are general enough to have a broad applicability, but specific enough to be consistently reliable over a large number of texts. However, there are *many challenges* to overcome. A major one that we have already cited is the lack of sufficient training examples that are labeled with a particular class. Natural language is very varied, capturing all possible variations in the examples and having sufficient overlap in the examples to discriminate good patterns from noisy patterns is almost impossible. We also expect the feature values to be sometimes inaccurate due to errors in the preprocessing phase (e.g., syntactic analysis) and to errors of human annotation of the training set. In addition, the number of potential features is very large, but only few of them are active in each example, and only a small fraction of them are relevant to the target concept. Moreover, the individual features and their values are

often ambiguous markers of several classes; in combination with other features they might become more discriminative. But, introducing more features might not necessarily reduce ambiguity as they themselves are often sources of ambiguity. This situation poses problems both for supervised and unsupervised learning.

When information extraction is performed in real time, extraction algorithms need to perform fast computations and their computational complexity should be taken an eye on.

4.3 The Classification Scheme

A *classification scheme* describes the semantic distinctions that we want to assign to the information units and to the semantic relations between these units. The set can have the form of a straight list, for instance, when we define a list of named entity classes to be identified in a corpus (e.g., the classes **protein**, **gene**, **drug**, **disease** of information in biomedical texts). Or, the scheme can be characterized by its own internal structure. It might represent the labels that can be assigned to entities or processes (the entity classes), the attribute labels of the entity classes, the subclasses and the semantic relations that might hold between instances of the classes, yielding a real semantic network. For instance, in texts of the biomedical domain one might be interested in the **protein** and **gene** subclasses, in the protein attribute **composition** or in the relation **is located on** between a protein and a gene. In addition, this scheme preferably also integrates the constraints on the allowable combinations and dependencies of the semantic labels.

Semantic labels range from generic labels to domain specific labels. For instance, the semantic roles **sayer** in a **verbal process** and **verbiage** in a verbal process are rather generic information classes, while **neurotransmitter** and **ribonuclear inclusion** are quite domain specific. One can define all kinds of semantic labels to be assigned to information found in a text that is useful in subsequent information processing tasks such as information retrieval, text summarization, data mining, etc. Their definition often relies on existing taxonomies that are drafted based on linguistic or cognitive theories or on natural relationships that exist between entities. In case of a domain specific framework of semantic concepts and their relations we often use the term *ontology*.

In this book we are mostly interested in semantic labels that can be used for open domain tasks and more specifically open domain information retrieval. To accomplish such tasks, a semantic annotation of the text constituents preferably identifies at an *intra-clause or -sentence level*:

- 1) The type of action or state associated with the verb, possibly expressed in terms of primitive actions and states;
- 2) The entities participating in the action or state (normally expressed as arguments);
- 3) The semantic role of the participants in the action or state;
- 4) Possibly a more fine grained characterization of the type of the entity (e.g., person, organization, animal, ...);
- 5) Coreferent relationships between noun phrase entities;
- 6) Temporal expressions;
- 7) Spatial expressions.

Coreferent relations are also found *across clauses, sentences* and even *documents*. In a more advanced setting, information extraction can detect temporal and spatial relations within and across documents.

If information extraction is done in a specific domain with a specific task in mind, then we refine the label set for entities and their relations. For instance, in the domain of natural disasters, labels such as the **number of victims**, the **numbers of houses destroyed**, etc... might be useful to extract. In a business domain it might be interesting to extract the **price of a product**, the **e-mail of a company's information desk** or the **company a person works for**. In the legal domain it is interesting to extract the **sentence** in a criminal case.

The output of a *low-level* semantic classification can become a feature in a *higher-level* classification. For instance, a list of relations attributed to a person entity might trigger the concept **restaurant visit** by that person.

In the following sections and chapters we focus on information extraction approaches and algorithms that have proven their usefulness in extracting both semantic information that is labeled with generic and rather abstract classes, and domain specific information.

4.4 The Information Units to Extract

Our next question is what information units or elements we want to identify, classify and eventually extract from the texts. This process is often referred to as segmentation (Abney, 1991). When we use these information units in the indices of the texts, we call them text regions. The smallest textual units to which meaning is assigned and thus could function as an information unit are the *free morphemes* or *root forms* of words. However, some words on their own do not carry much meaning, but have functional properties in the syntactic structure of a text. These function words alone

can never function as information units. Single words, base phrases or chunks, larger phrases, clauses, sentences, passages or structured document parts (e.g., section or chapters) might all be considered as information units to extract.

The extraction units most commonly used in information extraction are *base phrases* (e.g., base noun and verb phrases). A base noun phrase or noun chunk in English can be defined as a maximal contiguous sequence of tokens in a clause whose POS tags are from the set {JJ, VBN, VBG, POS, NN, NNS, NNP, NNPS, CD}.¹ A base verb phrase is a maximal contiguous sequence of tokens in a clause whose POS tags are from the set {VB, VBD, VBP, VBZ} possibly combined with a tag from the set {VBN, VBG}.²

One could define within a base noun phrase *nested noun phrases*. Here we have to deal with possessive noun phrases (e.g., **her** promotion, **John's** book) and modifier noun phrases or prenominal phrases (e.g., **student** scholarship, **University** officials). These noun phrases are still easy to detect in English texts. On the other hand a base noun phrase can be augmented with modifiers headed by a preposition (e.g., **Massachusetts Institute of Technology**). For this task we need a syntactical parser that captures the syntactic dependency structure of each sentence in order to distinguish a noun phrase that modifies another noun phrase from one that modifies a verb phrase (e.g., leaving **my house** in a hurry and leaving **my house in my daddy's neighborhood**). The detection of verb phrases and their arguments also requires a syntactic parse.

Although we have the tools to identify individual nouns and verbs, base phrases and full phrases, it is sometimes difficult to define which format is best suited to delimit an entity or the process it is involved in (e.g., **Massachusetts Institute of Technology** versus **Rik De Busser of Leuven**). This problem is especially significant in the biomedical domain (see Chap. 9). It can partially be solved by learning collocations, i.e., detecting words that co-occur together more often than by chance in a training corpus by means of statistical techniques (e.g., mutual information statistic, chi-square statistic, likelihood ratio for a binomial distribution) (Dunning 1993; Manning and Schütze, 1999). With these techniques it is possible to learn an

¹ Penn Treebank tag set: JJ = adjective; JJR = adjective, comparative; JJS = adjective, superlative; VBN = verb, past participle; VBG = verb, gerund/present participle; POS = possessive ending; NN = noun, singular; NNP = proper noun, singular; NNS = noun, plural; NNPS = proper noun, plural; CD = cardinal number.

² VB = verb, base form; VBD = verb, past tense; VBP = verb, non-3rd person singular present; VBZ = verb, 3rd person singular present.

expression (e.g., a noun phrase) consisting of two or more words that corresponds to some conventional way of saying things. Usually, the collocated words found add an element of meaning that cannot be predicted from the meanings of their composing parts.

It is also possible to consider all candidate phrases in an information extraction task (e.g., **the university student of Bulgaria**: consider: **the university student of Bulgaria, the university student, the student of Bulgaria, the student**) and to select the one among the candidates that belongs to a certain semantic class with a large probability. For instance, in a noun phrase coreference resolution task, such an approach has been implemented. Boundary detection and classification of the information unit are sometimes seen as two separate tasks, each relying on a different feature set. A difficult problem to deal with and that is comparable with the nested noun phrase problem regards information units that are conjunctions of several individual units. Here too, all different possibilities of phrases can be considered.

Not only basic noun and verb phrases are identified, individual words or expressions might be useful to classify, such as certain adverbs and adverbial expressions (e.g., **today, up to here**).

We also consider information units that extend phrase boundaries such as the classification of sentences or passages. For such larger units we cross the domain of text categorization. The semantic classifications described in this book offer valuable features to classify larger text units with semantic concepts, and the technologies discussed can be used to classify relationships between clauses, sentences and passages (e.g., to detect rhetorical and temporal relationships) that are very valuable when semantically classifying a passage (e.g., classifying the passage as a **visit to the dentist**; or classifying it as a **procedure**).

4.5 The Features

Machine learning approaches rely on feature vectors built from a labeled (already classified) or an unlabeled document collection. Depending upon the classification task a *set of features is selected*. We usually do not use all features that are present in a text, but select a number of important ones for the information extraction task at hand in order to reduce the computational complexity of the training of the classifier, and at the same time we keep as much as possible class discriminatory information. In the framework of an open domain information extraction task, it is important that

the features are generic enough to be used across different domains and that their values can automatically be detected.

The information units that we have identified in the previous section are described with certain features, the values of which are stored in the feature vector of the unit that is semantically classified. The features themselves can be classified in different types. Features can have numeric values, i.e., discrete or real values. A special discrete value type is the Boolean one (i.e., value of 1 or 0). Features can also have nominal values (e.g., certain words), ordinal values (e.g., the values 0 = small number, 1 = medium number; 2 = large number), or interval or ratio scaled values. We can make conversions to other types of features. For instance, a feature with nominal values can be translated to a number of features that have a Boolean or real value (e.g., if the value of a feature represents a word in a vocabulary, the feature can be translated into a set of features, one for each word in the vocabulary, which is advantageous, if one wants to give the words a weight).

Features can also be distinguished by their position in the text. First, we can define features that occur in the *information unit* itself, such as the composition of letters and digits of an entity name. Secondly, there are the features that occur in the *close neighborhood* or *context window* of the token string to be classified. In this category there are the features of the words that surround an information unit to be classified. Thirdly, if a relationship between two entities is to be found, features *that are linked with each of the entity or with both entities can be defined*. Fourth, the broader context in which the information unit occurs can give additional evidence for its semantic classification. In this case it is convenient to define features that occur in the *complete document* or *document collection*. For instance, when classifying an entity name in a sentence, we might rely on the assumption of one sense per discourse (Yarowski, 1995). Thus, repetitions of the name or reliably resolved acronyms or abbreviations of the name can offer additional context and evidence to classify the entity name (Chieu and Ng, 2002). Analogically, in a relation extraction task when we have first resolved the noun phrases that refer to the same entity, we can define features that are selected from different documents in order to learn the relation between two entity names.

In the next section we discuss the most commonly used features in typical information extraction tasks. We classify the features in lexical, syntactic, semantic and discourse features. The features, their types and their values are illustrated in tables that explicitly group the features used in an extraction task. In this way we give the implementer of an information extraction system two views on the feature selection process. On one hand, the distinction in lexical, syntactic, semantic and discourse features groups the typical methodologies and feature selection algorithms needed

for the text analysis. On the other hand illustrative tables summarize feature selection for a particular extraction task. For a particular feature that is cited in these tables, we give its most common value type.

1. The features for a named entity recognition task are based on the work of Bikel et al. (1999), Borthwick (1999), Collins and Singer (1999), Zhou and Su (2002), and Bunescu and Mooney (2004) (Table 4.1). In named entity recognition features typical for the entity name itself and contextual features play a role.
2. The features for the single-document noun phrase coreference resolution task refer to the work of Cardie and Wagstaff (1999), Soon et al. (2001) and Müller et al. (2002) (Table 4.2). Most reference resolution programs determine the relationship between a noun phrase and its referent only from the properties of the pair. The context of both noun phrases is usually ignored.
3. The features for the cross-document coreference resolution refer to the work of Bagga and Baldwin (1998), Gooi and Allan (2004) and Li et al. (2004) (Table 4.3). Cross-document noun phrase coreference resolution is per se a word sense disambiguation task. Two names refer to the same entity if their contexts in the different documents sufficiently match. Especially, proper names in these contexts are indicative of the meaning of the target proper name. Often, cross-document coreference resolution relies on single-document coreference resolution for solving the coreferents in one text, and it uses cross-document resolution for disambiguating identical names across texts, although mixed approaches that combine both tasks are also possible.
4. The features for a semantic role recognition task rely on the work of Fleischman and Hovy (2003), Pradhan et al. (2004), Mehay et al. (2005) (Table 4.4). Syntactic and structural features (e.g., position) play an important role besides some lexical characteristics (e.g., use of certain prepositions).
5. In relation recognition our features are based on the work of Hasegawa et al. (2004) (Table 4.5). In this task contextual features are quite important: There is no way to be certain that the sentence **He succeeds Mr. Adams** is a corporate management succession. It may refer to a political appointment, which is considered irrelevant, if we want to identify management successions. A large window of context words is here advisable for feature selection.
6. The features used to detect temporal expressions or timexes were previously described in Mani (2003) and Ahn et al. (2005) (Table 4.6). Processing of temporal information regards the detection and possible normalization of temporal expressions in text; their classification in

absolute and relative expressions and in case of the latter the computation of the absolute value, if possible; and the ordering of the expressions in time (Mani et al., 2005).

The feature set used in information extraction is very rich and varied. Natural language data is a domain that particularly benefits from rich and overlapping feature representations.

Quite often feature values are transformed when used in an information extraction task. For instance, one can aggregate a number of different feature values by one general feature value. This process is referred to as *feature extraction* or feature generation. An example of feature extraction is when semantic classifications of words are used as features in complex extraction tasks (see *infra*).

4.5.1 Lexical Features

Lexical features refer to the attributes of lexical items or words of a text. One can make a distinction between the words of the information unit that is to be classified, and its context words.

Table 4.1. Typical features in a named entity recognition task of the candidate entity name i that occur in the context window of l words.

FEATURE	VALUE TYPE	VALUE
Short type	Boolean	True if i matches the short type j ; False otherwise.
POS	Nominal	Part-of-speech tag of the syntactic head of i .
Context word	Boolean or real value between 0 and 1; Or nominal.	True if the context word j occurs in the context of i ; False otherwise; If a real value is used, it indicates the weight of the context word j . Alternatively, the context word feature can be represented as one feature with nominal values.
POS left	Nominal	POS tag of a word that occurs to the left of i .
POS right	Nominal	POS tag of a word that occurs to the right of i .
Morphological prefixes/suffixes	Nominal	Prefix or suffix of i .

In named entity recognition tasks morphological characteristics of the information to be classified is often important. By morphological characteristics we mean the occurrence of specific character conventions such as the occurrence pattern of digits and capital letters in a word or sequence of words. Because it is difficult to represent all possible compositions in a feature vector, entities are often mapped to a restricted number of feature templates that are a priori defined and are sometimes called *short types* (Collins, 2002). A *short type* of a word can, for instance, be defined by replacing any maximal contiguous sequence of capital letters with ‘A’, of lowercase letters with ‘a’ and of digits with ‘0’, while keeping the other non alpha-numeric characters. For example, the word **TGV-3** would be mapped to **A-0**. It is also possible to define short types for multi-word expressions. A template can also represent more refined patterns (e.g., the word contains one digit at a certain position or contains a digit and a period at a certain position).

Simple heuristic rules allow detecting certain attributes of an information unit. For instance, the title, first name, middle name and last name of a person can be identified and used as a feature in coreference resolution.

It is common that words or compound terms have different *variant spellings*, i.e., an entity can have different mentions. Especially, proper names such as person names can occur in a text in different alias forms. Although the task of alias recognition in itself is a noun phrase coreference resolution task, often a simple form of alias recognition is a priori applied yielding classification features such as “is alias” and “is weak alias”. They especially aim at detecting variations concerning punctuation (e.g., **USA** versus **U.S.A**), capitalization (e.g., **Citibank** versus **CITIBANK**), spacing (e.g., **J.C. Penny** versus **J. C. Penny**), abbreviations and acronyms (e.g., **information retrieval** versus **IR**), misspellings including omissions (e.g., **Collin** versus **Colin**), additions (**McKeown** versus **MacKeown**), substitutions (e.g., **Kily** versus **Kyly**), and letter reversals (e.g., **Pierce** versus **Peirce**). Punctuation and capitalization variations can be resolved – although not in an error-free way - by simple normalization. Abbreviations and acronyms can be normalized by using a translation table of abbreviations or acronyms and their corresponding expansions. Or, simple rules for acronym resolution might be defined. Especially for detecting misspelling, edit distances are computed. Then the similarity between two character strings is based on the cost associated with converting one pattern to the other. If the strings are of the same length, the cost is directly related to the number of symbols that have been changed in one of the strings so that the other string results. In the other case, when the strings have a different length, characters have to be either deleted or inserted at certain places of the test string. The edit distance $D(A,B)$ is defined as the minimum total number of

(possibly weighted) substitutes S , insertions I , and deletions R required to change pattern A into pattern B :

$$D(A,B) = \min_j [S(j) + I(j) + R(j)] \quad (4.2)$$

where j runs over all possible combinations of symbol variations in order to obtain B from A . Dynamic programming algorithms are usually used to efficiently obtain B from A (Skiena, 1998, p. 60 ff.).

Another alias detection heuristic refers to the matching of strings except for articles and demonstrative pronouns. An evaluation of different techniques for proper name alias detection can be found in Branting (2003). The first mention of the entity in a text is usually taken as the most representative. It is clear that alias resolution across different documents requires additional context matching as names that are (slightly) differently spelled might refer to different entities.

It is also common in text that entities are referred to by their *synonym*, *hypernym*, *hyponym* or sometimes *meronym*. A synonym is a term with the same meaning as the source term, but differently spelled. A hypernym denotes a more general term, while a hyponym refers to a more specific term compared to the source term. A meronym stands for a part of relation. Thesauri or lexical databases such as WordNet (Miller, 1990) usually contain these term relationships. It is not always easy to correctly detect synonyms, hypernyms and hyponyms in texts because of the different meanings that words have. The lexica often cite the different meanings of a word, but sometimes lack sufficient context descriptions for each meaning in order to easily disambiguate a word in a text.

Other lexical features regard *gender* and *number* of the information unit, or of the head of the unit if it is composed of different words. They are, for instance, used as matching features in a noun phrase coreference task. An entity can have as gender: Masculine, feminine, both masculine and feminine and neutral. Additional knowledge of the gender of persons is helpful. It could be detected by relying on lists of first names in a particular language or culture that are classified according to gender, when the person is mentioned with his or her first name and when the first name does not have an ambiguous gender (e.g., **Dominique** in French). The form of addressing a person also acts as a cue in determining a person's gender (e.g., **Mrs.** Foster). For common nouns, we have to infer the gender from additional knowledge sources. Number information is usually provided by the part-of-speech tagger where a tag such as **NNS** refers to a plural noun.

Table 4.2. Typical features in a single-document noun phrase coreference resolution task of the syntactic heads, i and j , of two candidate coreferent noun phrases in text T where $i < j$ in terms of word position in T .

FEATURE	VALUE TYPE	VALUE
Number agreement	Boolean	True if i and j agree in number; False otherwise.
Gender agreement	Boolean	True if i and j agree in gender; False otherwise.
Alias	Boolean	True if i is an alias of j or vice versa; False otherwise.
Weak alias	Boolean	True if i is a substring of j or vice versa; False otherwise.
POS match	Boolean	True if the POS tag of i and j match; False otherwise.
Pronoun i	Boolean	True if i is a pronoun; False otherwise.
Pronoun j	Boolean	True if j is a pronoun; False otherwise.
Appositive	Boolean	True if j is the appositive of i ; False otherwise.
Definiteness	Boolean	True if j is preceded by the article “the” or a demonstrative pronoun; False otherwise.
Grammatical role	Boolean	True if the grammatical role of i and j match; False otherwise.
Proper names	Boolean	True if i and j are both proper names; False otherwise.
Named entity class	Boolean	True if i and j have the same semantic class (e.g., person, company, location); False otherwise.
Discourse distance	Integer ≥ 0	Number of sentences or words that i and j are apart.

In many semantic classifications the context words are very important. The size of the window with context words usually varies according to the extraction task. In named entity recognition the window size is usually quite small (two or three words on the left or the right of the target word yielding a window of respectively of 5 or 7 words). In a cross-document coreferent resolution task, the window can be quite large (e.g., 50 words, or the sentence in which the target word occurs). Words in context windows might receive a weight that indicates their importance. Quite often classical weighting functions such as $tf \times idf$ are used for this purpose. The term frequency (tf) is valuable when the words of different context windows are combined in one vector. This is, for instance, the case when in

Table 4.3. Typical features in a cross-document noun phrase coreference resolution task of the syntactic heads, i and j , of two candidate coreferent noun phrases where i and j occur in different documents.

FEATURE	TYPE	VALUE
Context word	Boolean or real value between 0 and 1	True if the context word k occurs in the context of i and j ; False otherwise; If a real value is used, it indicates the weight of the context word; Proper names, time and location expressions in the context might receive a high weight.
Named entity class	Boolean	True if i and j have the same semantic class (e.g., person, company, location); False otherwise.
Semantic role	Boolean	True if the semantic role of i matches the semantic role of j ; False otherwise.

one document the context windows of identical or alias mentions of an entity can be merged while relying on the one sense per discourse principle which, for instance, for proper names can be accepted with high accuracy. The term frequency is then computed as the number of times a term occurs in the window(s). The inverse document frequency (*idf*) is useful to demote term weights when the term is a common term in the document collection under consideration or in a reference corpus in the language of the document. The *idf* of term i is usually computed as $\log(N/n_i)$ where N is the number of documents in the collection and n_i the number of documents in the collection in which i occurs. In context windows, stop words or function words might be neglected. For certain tasks such as cross-document noun phrase coreference resolution, proper names, time and location expressions in the context might receive a high weight. In order to find coreferring names across documents, the semantic roles and processes in which the entities are involved can yield additional cues.

4.5.2 Syntactic Features

The most common syntactic feature used in information extraction is the *part-of-speech* (POS) of a word. Part-of-speech taggers that operate with a

very high accuracy are commonly available. The part-of-speech of a word often plays a role in determining the values of other features.

So, for instance the *definiteness of an information unit* or noun phrase entity can be approximately defined if the unit is preceded by the article “the” or a demonstrative pronoun (e.g., I saw a man and **the man** was old. **That person** wore strange clothes). In this example **A man** refers to indefinite information. Defining definiteness is valuable to detect anaphoric noun phrase coreferents in texts (Yang et al., 2004). Definite noun phrases usually refer to content that is already familiar or to content items of which there exist only one (e.g., the U.S.). Definiteness can be split up in two separate Boolean features: Definite and indefinite (Ng and Cardie, 2002), which allows describing cases that are neither definite nor indefinite.

Alias recognition or weak alias recognition (cf. supra) can also rely on part-of-speech tags. The part-of-speech tag gives us information on words that might be removed for string matching of the candidate aliases. For instance for proper names, we can remove words that do not have the part-of-speech **NNP** (single proper name) or **NNPS** (plural proper name). For words that belong to the general part-of-speech type **NN** (noun), especially the head noun is important in the matching of candidate aliases.

Detecting the *type of phrase* (e.g., a noun phrase such as **the big bear**, a prepositional noun phrase such as **in the cold country**) is important in a semantic role recognition task. The syntactic head of a phrase is here a useful feature. The *syntactic head* of a phrase is the word by whose part-of-speech the phrase is classified (e.g., **man** in the noun phrase: **the big man**). In timex recognition, the following information units are usually considered as candidates: Noun, noun phrase, adjective, adverb, adjective phrase and adverb phrase.

The *voice of a clause* (i.e., passive or active) is a useful feature in a relation extraction task. It can be detected based on surface expressions in the texts and the part-of-speech of the verb words. Another mode feature determines whether the sentence is affirmative or negative. This feature is more difficult to accurately detect.

A number of syntactic features rely on a *parsing of the sentence’s structure*. Unfortunately, sentence parsers are not available for every language. The grammatical role of a phrase in a sentence or clause such as subject, direct object and indirect object might play a role in the extraction process. Grammatical roles, which are sometimes also called syntactic roles, are detected with the help of rules applied on the parse tree of a sentence. In certain languages the grammatical role of nouns and pronouns can be detected by their morphological form that indicates cases such as nominative, accusative, genitive and ablative. The grammatical role is important in a

coreference resolution task as antecedent and referent often match with regard to their grammatical role (Yang et al., 2004).

Parse information is also important in detecting relations between entities (Culotta and Sorensen, 2004). For instance, defining whether the two noun phrase entities are in a modifier relation, or defining their grammatical roles in the sentence acts as a useful feature in relation recognition.

4.5.3 Semantic Features

Semantic features refer to semantic classifications of single- or multi-word information units. The *semantic features* act as features in other semantic classification tasks. An example is **John Barry works for IBM** where **John Barry** and **IBM** are already classified respectively as person name and company name. These more general features are then used in the recognition of the relation **works for**. There are multiple circumstances

Table 4.4. Common features in a generic semantic role recognition task of clause constituent *i*.

FEATURE	VALUE TYPE	VALUE
Phrase type	Nominal	Phrase type (e.g., noun phrase, verb phrase) as determined by the POS tag of the syntactic head of <i>i</i> .
Syntactic head	Nominal	The word that composes the syntactic head of the phrase that represents <i>i</i> .
Grammatical role	Nominal	The grammatical role of <i>i</i> .
Voice	Nominal	The voice of the clause of which <i>i</i> is part: Active or passive.
Named entity class	Nominal	Name of the named entity class (e.g., person, organization) of the syntactic head of <i>i</i> ; Undefined when <i>i</i> is not a noun phrase.
Relative distance and position	Integer	The relative distance of the syntactic head of <i>i</i> with regard to the process can be defined as a number that is proportional with the distance (e.g., in terms of words); The numbering (e.g., negative or positive) provides also the distinction whether <i>i</i> occurs before or after the process in the clause; Is zero when <i>i</i> represents the process in the clause.

Table 4.5. Common features in a relation recognition task between two noun phrase entities i and j in a clause c considering a context of l words.

FEATURE	VALUE TYPE	VALUE
Context word	Boolean or real value between 0 and 1; Or nominal.	True if the context word k occurs in the context of i and j ; False otherwise; If a real value is used, it indicates the weight of the context word k . Alternatively, the context word feature can be represented as one feature with nominal values.
POS context word	Nominal	For each context word, there is a feature that designates the word's POS tag.
Semantic role i	Nominal	Semantic role of phrase i ; Undefined when i is a modifier.
Semantic role j	Nominal	Semantic role of phrase j ; Undefined when j is a modifier.
Modifier i	Boolean	True if i is a modifier of j ; False otherwise.
Modifier j	Boolean	True if j is a modifier of i ; False otherwise.
Affirmative	Boolean	True if the clause c in which i and j occur is affirmative; False otherwise.

where the replacement of words and terms by more general semantic concepts is advantageous especially when the features are used to semantically classify larger information units or in more complex classification tasks such as coreference resolution. In coreference resolution it is very important to use semantic classes such as **female**, **male**, **person** and **organization**, or **animate** and **inanimate** and to find agreement of antecedent and referent on these classes. Semantic features may involve simple identification of the name of a day or month by respectively the classes **day** or **month**, the recognition of useful categories such as **person name**, **company name**, **number** and **money**, and the recognition of very general classes such the **sayer** in a **verbal** process.

An additional advantage is that semantic tagging of individual words enables rules of greater generality than rules based exclusively on exact words. In this way it offers a solution to problems caused by the sparseness of training data and the variety of natural language expressions found in texts.

There are several ways for identifying the semantic features. Firstly, they can be detected with the typical information extraction techniques described in this book, such as named entity recognition and semantic role recognition. Secondly, we can rely on external knowledge sources that are in the form of machine-readable dictionaries or lexica, which can be general or domain specific. Especially useful is a semantic lexicon that can be

used to tag individual words with semantic classes appropriate to the domain. Semantic class determination relying on general lexical databases such as WordNet (Miller, 1990) is not easy when they lack the necessary contextual expressions to disambiguate word meanings. There also exist gazetteers that contain geographical or other names. In addition, semantic lexica might be incomplete and in practical applications generic resources often have to be complemented with domain specific resources. A list of the most common first or last names can be used in a named entity recognition task (e.g., US Census list of the most common first and last names in the US).

4.5.4 Discourse Features

Discourse features refer to features the values of which are computed by using text fragments, i.e., a discourse or a connected speech or writing, larger than the sentence. Many discourse features are interesting in an information extraction context.

A very simple example is *discourse distance*. In relation recognition the distance between two entities is often important as it is assumed that distance is inversely proportional with semantic relatedness. Especially in single-document coreference resolution discourse distance is relevant. Discourse distance can be expressed by the number of intervening words or by the number of intervening sentences.

Discourse features such as *rhetorical*, *temporal* and *spatial relations* between certain information found in the texts are important in the semantic classification of larger text units. For instance, the temporal order of cer-

Table 4.6. Common features of phrase i in a timex recognition task considering a context window of l words.

FEATURE	TYPE	VALUE
Context word	Boolean or real value between 0 and 1; Or nominal.	True if the context word j occurs in the context of i ; False otherwise; If a real value is used, it indicates the weight of the context word j . Alternatively, the context word feature can be represented as one feature with nominal values.
Short type	Boolean	True if i matches the short type j ; False otherwise.

tain actions is a significant indicator of script based concepts expressed in texts (e.g., a restaurant visit, a bank robbery). The recognition of temporal expressions (timexes), their possible anchoring to absolute time values and their relative ordering are themselves considered as information extraction tasks (e.g., Mani et al., 2005; Mani, 2003). TimeML (Pustejovsky et al., in Mani et al., 2005) is a proposed metadata standard for markup of events and their temporal anchoring in documents. The drafting of classification schemes of temporal relationships goes back to Allen (1984) (e.g., **before**, **after**, **overlaps**, **during**, etc...). More recent ontological classification schemes aim to logically describe the temporal content of Web pages and to make inferences or computations with them (Hobbs and Pan 2004). Experiments with regard to the automatic classification of temporal relationships are very limited (Mani et al., 2003) and few studies report on adequate discourse features except for features that track shifts in tense and aspect. This is why we did not include a separate table for typical temporal relationship features.

4.6 Conclusions

Information extraction is considered as a pattern classification task. The candidate information unit to be extracted or semantically classified is described by a number of features. The feature set is very varied. However, a number of generic procedures are used in feature selection and extraction. They comprise lexical analysis, part-of-speech tagging and possibly parsing of the sentences. These primitive procedures allow identifying a set of useful information extraction features that can be found in open and closed domain document collections. Discourse features are used to a lesser extent, but will certainly become more important in future semantic classifications. Elementary information classifications, such as named entity recognition, yield semantic features that can be used in more complex semantic classifications, such as coreference resolution and relation recognition. The results of entity relation and time line recognition tasks can in their turn act as features in a script recognition task. Such an approach, to which we refer as a cascaded model, starts from semantically classifying small information units, and in a kind of bootstrapping way uses these to classify larger information units. This model opens avenues for novel learning algorithms and could yield semantic representations of texts at various levels of detail.

In the following two chapters we discuss the typical learning algorithms used in information extraction.

4.7 Bibliography

- Abney, Steven P. (1991). Parsing by chunks. In Steven P. Abney, Robert C. Berwick and Carol Tenny (Eds.), *Principle Based Parsing: Computation and Psycholinguistics* (pp. 257-278). Dordrecht, The Netherlands: Kluwer.
- Ahn, David, Sisay F. Adafre and Maarten de Rijke (2005). Extracting temporal information from open domain text. In *Proceedings of the 5th Dutch-Belgian Information Retrieval Workshop (DIR'05)*.
- Allen, James (1984). Towards a general theory of action and time. *Artificial Intelligence*, 23 (2), 123-154.
- Bagga, Amit and Breck Baldwin (1998). Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL'98)* (pp. 79-85). Morgan Kaufmann: ACL.
- Bikel, Daniel M., Richard Schwartz and Ralph M. Weischedel (1999). An algorithm that learns what's in a name. *Machine Learning*, 34 (1/2/3), 211-231.
- Borthwick, Andrew E. (1999). *A Maximum Entropy Approach to Named Entity Recognition*. Ph.D. thesis, Computer Science Department, New York University.
- Branting, Karl L. (2003). A comparative evaluation of name matching algorithms. In *Proceedings of the 9th International Conference on Artificial Intelligence and Law* (pp. 224-232). New York: ACM.
- Bunescu, Razvan and Raymond J. Mooney (2004). Collective information extraction with relational Markov networks. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (pp. 439-446). East Stroudsburg, PA: ACL.
- Cardie, Claire and Kiri Wagstaff (1999). Noun phrase coreference as clustering. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Very Large Corpora* (pp. 82-89). San Francisco, CA: Morgan Kaufmann.
- Chieu, Hai L. and Hwee T. Ng (2002). Named entity recognition: A maximum entropy approach using global information. In *COLING 2002. Proceedings of the 19th International Conference on Computational Linguistics* (pp. 190-196). San Francisco: Morgan Kaufmann.
- Collins, Michael (2002). Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics* (pp. 489-496). San Francisco: Morgan Kauffman.
- Collins, Michael and Yoram Singer (1999). Unsupervised models for named entity classification. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, College Park, MD.
- Craven, M., et al. (2000). Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118, 69-113.
- Culotta, Aron and Jeffrey Sorenson (2004). Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (pp. 424-430). East Stroudsburg, PA: ACL.

- Dunning, Ted (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 61-74.
- Fleischman, Michael and Eduard Hovy (2003). A maximum entropy approach to FrameNet tagging. In *Proceedings of the Human Language Technology Conference of the North American Chapter for Computational Linguistics*. East Stroudsburg, PA: ACL.
- Gooi, Chung Heong and James Allan (2004). Cross-document coreference on a large scale corpus. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 9-16). East Stroudsburg, PA: ACL.
- Graesser, Arthur C. and Leslie F. Clark (1985). *Structures and Procedures of Implicit Knowledge (Advances in Discourse Processes XVII)*. Norwood, NJ: Ablex Publishing Corporation.
- Hasegawa, Takaaki, Satoshi Sekine and Ralph Grishman (2004). Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (pp. 416-423). East Stroudsburg, PA: ACL.
- Hobbs, Jerry R. and Feng Pan (2004). An ontology of time for the semantic Web. *ACM Transactions on Asian Language Information Processing*, 3 (1), 66-85.
- Jones, William P. and George W. Furnas. (1987). Pictures of relevance: A geometric analysis of similarity measures. *Journal of the American Society for Information Science*, 38 (6), 420-442.
- Li, Xin, Paul Morie and Dan Roth (2004). Robust reading: Identification and tracing of ambiguous names. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 17-24). East Stroudsburg, PA: ACL.
- Mani, Inderjeet (2003). Recent developments in temporal extraction. In Nicolas Nicolov and Ruslan Mitkov (Eds.), *Proceedings of RANLP'03*. Amsterdam: John Benjamins.
- Mani, Inderjeet, James Pustejovski and Robert Gaizauskas (Eds.) (2005). *The Language of Time: A Reader*. Oxford, UK: Oxford University Press.
- Mani, Inderjeet, Barry Schiffman and Jianping Zhang (2003). Inferring temporal ordering of events in news. In *Proceedings of the Human Language Technology Conference (HLT-NAACL'03)* (pp. 55-57). Edmonton, CA.
- Manning, Christopher D. and Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Boston, MA: The MIT Press.
- Mehay Dennis N., Rik De Busser and Marie-Francine Moens (2005). Labeling generic semantic roles. In Harry Bunt, Jeroen Geertzen and Elias Thyse (Eds.), *Proceedings of the Sixth International Workshop on Computational Semantics (IWCS-6)* (pp. 175-187). Tilburg, The Netherlands: Tilburg University.
- Miller, George A. (Ed.) (1990). Special issue: WordNet: An on-line lexical database. *International Journal of Lexicography*, 3 (4).
- Müller, Christoph, Stefan Rapp and Michael Strubbe (2002). Applying co-training to reference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 352-359). San Francisco: Morgan Kaufmann.
- Ng, Vincent and Claire Cardie (2002). Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Asso-*

- ciation for Computational Linguistics* (pp. 104-111). San Francisco: Morgan Kaufmann.
- Pradhan, Sameer, Wayne Ward, Kadri Hacioglu, James H. Martin and Dan Jurafsky (2004). Shallow semantic parsing using support vector machines. In *Proceedings of the Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL 2004)*. East Stroudsburg, PA: ACL.
- Skiena, Steven S.K. (1998). *The Algorithm Design Manual*. New York, NY: Springer.
- Soon Wee Meng, Hwee Tou Ng and Daniel Lim (2001). A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics* 27(4), 2001, 521-544.
- Theodoridis, Sergios and Konstantinos Koutroumbas (2003). *Pattern Recognition*. Amsterdam, The Netherlands: Academic Press.
- Yang, Xiaofeng, Jian Su, Guodong Zhou and Chew Lim Tan (2004). Improving pronoun resolution by incorporating coreferential information of candidates. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (pp. 128-135). East Stroudsburg, PA: ACL.
- Yarowski, David (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33th Annual Meeting of the Association for Computational Linguistics* (pp. 189-196). Cambridge, MA.
- Zhou, GuoDong and Jian Su (2002). Named Entity Recognition using an HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 473-480). San Francisco, CA: Morgan Kaufmann.