Chapter 2
Data Annotation and Preprocessing

2.1 Data Acquisition

Data acquisition sources and methods are different for different text mining tasks. Considering the sources of data, there are usually two situations. The first is open domain data. For example, when building a system for mining public opinion from social media, the data naturally come from all available public social networks, including mobile terminals. Although the subject of the mined text may be limited to one or a few specific topics, the data source is open. The situation is closed domain data. For example, the data processed by text mining tasks oriented toward the financial field are proprietary data from banks and other financial industries; similarly, the texts processed by tasks oriented toward hospitals exist in a private network from the internal institutions of the hospital, and they cannot be obtained by public users. Of course, the so-called open domain and closed domain are not absolute, and when implementing a system in practice, it is often not sufficient to solely rely on the data in a specific domain because they will mainly contain professional domain knowledge, while much of the data associated with common sense exists in public texts. Therefore, closed data need to be supplemented with data obtained from public websites (including Wikipedia, Baidu Encyclopedia, etc.), textbooks, and professional literature. Relatively speaking, the data from public networks (especially social networks) contain more noise and ill-formed expressions, so it takes more time to clean and preprocess them.

The following is an example of how movie reviews are obtained to illustrate the general method of data acquisition.

Before acquiring data, one must first know which websites generally contain the required data. The website *IMDb*[1] provides users with comments on movies, and there are many links to movies on the web page, as shown in Fig. 2.1.

[1] https://www.imdb.com/.

C. Zong et al., *Text Data Mining*, https://doi.org/10.1007/978-981-16-0100-2_2

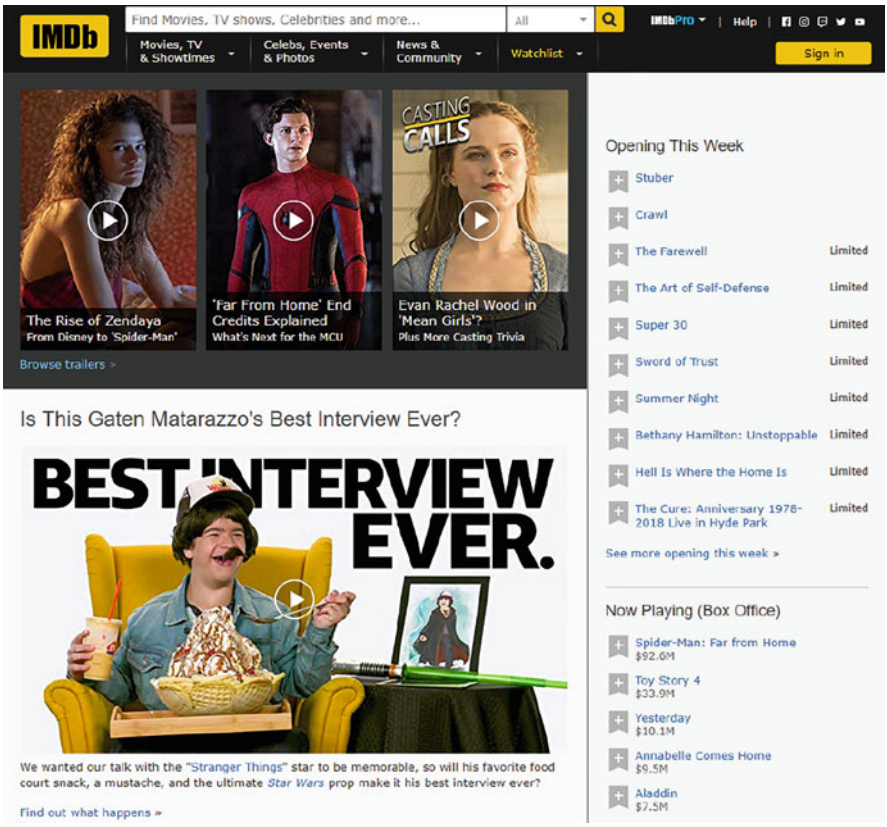

Fig. 2.1 The IMDb homepage

Taking "Mission: Impossible - Fallout" as an example, it can be seen that there are many comments on this film. As shown in Fig. 2.2, a total of 269,543 people gave comments, and the average score given is 7.7 (see the right top corner of Fig. 2.2).

A comment with its score is provided at the bottom of the main page belonging to the "Mission: Impossible - Fallout" film, as shown in Fig. 2.3, but this is not comprehensive. Click "See all 1,597 user reviews" in the bottom row of Fig. 2.3 to view the link to all comments in a comments page. At the end of the comments page, there is a "Load More" button. The user can click the button to obtain extra comments and download the data connected with a link by using Python's urllib2 library.

When using the Python programming language to crawl data from a website, the user must first check and then abide by the robot protocol of the website, which defines what website data can be crawled and what cannot be crawled. Figure 2.4 shows the robot protocol content of IMDb. The "Disallow" in the protocol limits the content that cannot be crawled (much of the search-related content cannot be

Fig. 2.2 The main page for "Mission: Impossible - Fallout"[2]

crawled). There is no restriction on the crawling of movie reviews, so it is legal and valid to crawl these contents, but the process must conform to the crawling time interval. This means that in the process of crawling, the speed should be reduced as much as possible. In fact, crawling process reflects access to the web server. If crawling makes too frequent requests, it will affect the operation of the web server. In addition, it is better to crawl a website when network traffic is low (e.g., at night) to avoid interfering with the normal operations of the website.

The data downloaded from the web page usually have a good structure. The `beautiful Soup` toolkit for Python can be employed to extract the content and obtain the links for the next page. When parsing a web page, the row delimiter ('\r', '\n') of the web page should be deleted, and there may be special symbols in the downloaded data such as " " and "<", which represent a space and the

[2] https://www.imdb.com/title/tt4912910/.

Fig. 2.3 The review page for "Mission: Impossible - Fallout"

less-than sign, respectively; these can be replaced if they are not necessary. Table 2.1 shows the corresponding meaning of some special symbols used in web pages.

After obtaining the comment content, it is necessary to clean the data and remove any noise words or text that is too short (which is usually meaningless). The processing procedure is specifically given as follows:

(1) Noise processing: There are likely to be English words or letters in downloaded Chinese text or symbols from other languages. This requires identification of the language type. The `Langdect` toolkit in Python can be used to help identify and delete those data that are not needed. In addition, the crawled microblog data may contain advertisement links, "@," and so on, which requires special handling. These links can be deleted directly, and the symbol "@" is usually followed by a user name, which can be determined by using simple methods such as rule-based or template-based approaches and then deleted.

(2) Conversion of traditional Chinese characters: There may be some traditional Chinese characters in the downloaded simplified Chinese text, which need to be converted into simplified characters. The conversion process can be performed with the help of the open source toolkit OpenCC[3] or other tools.

(3) Remove comments that are too short: For English comments, the number of words in the text can be directly counted by numerating spaces. However, for Chinese, Japanese, or other language text, it is necessary to segment the characters into words first before counting the number of words. If the length of a piece of text is shorter than a certain threshold (e.g., 5), it is usually removed.

[3] https://opencc.byvoid.com/.

```
# robts.txt for https://www.imdb.com properties
User-agent: *
Disallow:  /OnThisDay
Disallow:  /ads/
Disallow:  /ap/
Disallow:  /mymovies/
Disallow:  /r/
Disallow:  /register
Disallow:  /registration/
Disallow:  /search/name-text
Disallow:  /search/title-text
Disallow:  /find
Disallow:  /find$
Disallow:  /find/
Disallow:  /tvschedule
Disallow:  /updates
Disallow:  /watch/_ajax/option
Disallow:  /_json/video/mon
Disallow:  /_json/getAdsForMediaViewer/
Disallow:  /list/ls*/_ajax
Disallow:  /*/*/rg*/mediaviewer/rm*/tr
Disallow:  /*/rg*/mediaviewer/rm*/tr
Disallow:  /*/mediaviewer/*/tr
Disallow:  /title/tt*/mediaviewer/rm*/tr
Disallow:  /name/nm*/mediaviewer/rm*/tr
Disallow:  /gallery/rg*/mediaviewer/rm*/tr
Disallow:  /tr/
```

Fig. 2.4 The robot protocol of IMDb[4]

(4) The mappings of labels: Websites usually provide the labels for their categories, and the number of categories is potentially different from that of the predefined classifier employed, so it is necessary to map the labels or categories from one type to another. For example, the evaluation score in downloaded data uses a 5-point system, while the sentiment classifier can only distinguish the sentiment into two categories, positive and negative, so the samples with scores of 4 and 5 can be taken as positive samples, those with scores of 1 and 2 can be treated as negative samples, and the "neutral" samples with scores of 3 can be deleted. Certainly, if a classifier is trained with three categories, i.e., positive, neural, and negative, you would annotate those samples with a score of 3 as neutral and preserve the middle category.

[4] https://www.imdb.com/robots.txt.

Table 2.1 Corresponding table of special symbols used in web data[5]

Displayed result	Description	Entity name	Entity number
	Space		
<	Less than	&lt;	&#60;
>	Greater than	&gt;	&#62;
&	Ampersand	&amp;	&#38;
"	Quotation mark	&quot;	&#34;
'	Apostrophe	' (do not support IE)	&#39;
¢	Cent	&cent;	&#162;
£	Pound	&pound;	&#163;
¥	Yen	&yen;	&#165;
	Euro	&euro;	&#8384;
§	Section	&sect;	&#167;
©	Copyright	&copy;	&#169;
®	Registered trademark	&reg;	&#174;
™	Trademark	&trade;	&#8482;
×	Times sign	&times;	&#215;
÷	Division sign	&divide;	&#247;

The methods for acquiring open domain data for other tasks are very similar, but the annotation methods are different; for example, for automatic text summarization or information extraction, the annotation work is much more complicated than simply annotating categories.

2.2 Data Preprocessing

After data acquisition, it is usually necessary to further process the data. The main tasks include:

(1) Tokenization: This refers to a process of segmenting a given text into lexical units. Latin and all inflectional languages (e.g., English) naturally use spaces as word separators, so only a space or punctuation is required to realize lexicalization, but there are no word separation marks in written Chinese and some other agglutinative languages (e.g., Japanese, Korean, Vietnamese), so word segmentation is required first. This issue is mentioned above.

(2) Removing stop words: Stop words mainly refer to functional words, including auxiliary words, prepositions, conjunctions, modal words, and other high-frequency words that appear in various documents with little text information, such as *the*, *is*, *at*, *which*, *on*, and so on in English or 的(de), 了(le) and 是(shi)

[5] https://www.w3school.com.cn/html/html_entities.asp.

in Chinese. Although "是(be)" is not a functional word, it has no substantive meaning for the distinction of text because of its high frequency of occurrence, so it is usually treated as a stop word and removed. To reduce the storage space needed by the text mining system and improve its operating efficiency, stop words are automatically filtered out during the phase of representing text. In the process of implementation, a list of stop words is usually established, and all words in the list are directly deleted before features are extracted.

(3) **Word form normalization**: In the text mining task for Western languages, the different forms of a word need to be merged, i.e., word form normalization, to improve the efficiency of text processing and alleviate the problem of data sparsity caused by discrete feature representation. The process of word form normalization includes two concepts. One is lemmatization, which is the restoration of arbitrarily deformed words into original forms (capable of expressing complete semantics), such as the restoration of *cats* into *cat* or *did* into *do*. Another is stemming, which is the process of removing affixes to obtain roots (not necessarily capable of expressing complete semantics), such as *fisher* to *fish* and *effective* to *effect*.

The process of word form normalization is usually realized by rules or regular expressions. The Porter stemming algorithm is a widely used stemming algorithm for English that adopts a rule-based implementation method (Porter 1980). The algorithm mainly includes the following four steps: (a) dividing letters into vowels and consonants; (b) utilizing rules to process words with suffixes of `-s`, `-ing`, and `-ed`; (c) designing special rules to address complicated suffixes (e.g.,`-ational`, etc.); and (d) fine-tuning the processing results by rules. The basic process of the algorithm is presented in Fig. 2.5.

In the Porter stemming algorithm, only a portion of the main rewriting rules are given from Step 2 to Step 4, and the rest are not introduced individually, as this is simply an example to illustrate the basic ideas behind it.

The implementation code for the algorithm can be obtained from the following web page:

https://tartarus.org/martin/PorterStemmer/

In addition, the NLTK toolkit in Python also provides calling functions for the algorithm.

It should be noted that there is no uniform standard for stemming results, and different stemming algorithms for words in the same language may have different results. In addition to the Porter algorithm, the Lovins stemmer (Lovins 1968) and the Paice stemmer (Paice 1990) are also commonly used for English word stemming.

Algorithm The Porter Stemming Algorithm

Input: An English word;
Output: The stem or original type of input word;
Algorithm:
Step 1: Distinguishing vowels and consonants by using the following rules:
(1) Letters `a`, `e`, `i`, `o`, `u` are vowels;
(2) The letter `y` has the following three cases:
(a) If `y` is the beginning of a word, it is judged as a consonant. e.g., `y` is a consonant in the word `young`;
(b) If the previous letter of `y` is a vowel, `y` is judged as a consonant. e.g., `y` is a consonant in the word `boy`;
(c) If the previous letter of `y` is a consonant, `y` is judged as a vowel. e.g., `y` is a vowel in the word `fly`.
(3) All other letters except `a`, `e`, `i`, `o`, `u`, `y` are consonants.
Step 2: Processing words with `-s`, `-ing` and `-ed` suffixes by using the following rules:
(1) Words ending with `-s` are treated as follows:
(a) If the word ends with `-sses`, then restore it to `-ss`. e.g., the word `caresses` should be restored to `caress`;
(b) If the word ends with `-ies`, then delete `-es`. e.g., `cries` becomes `cri`;
(c) If the word ends with `-s` and one of all letters before `s` is a vowel at least, consider the following two cases:
(i) if the vowel is adjacent the last `s`, the word will not change. e.g., the word `gas` is the original type and does not need to change;
(ii) Otherwise, delete the last letter `s`. e.g., `gaps` restore to `gap`.
(2) If the word ends with `-ing` and the previous part of the word contains a vowel letter except for `ing`, delete `ing`. e.g., the word `doing` restore to `do`.
Step 3: Use the following rules to process words with other suffixes.
(1) If the word ends with `-y` and the previous part of `-y` contains vowel letters, `-y` is changed to `i`. e.g., the word `happy` is rewritten as `happi`.
(2) If the word ends with `-ational` and the previous section of `-ational` contains vowel letters, `-ational` is rewritten as `-ate`, for example, the word `relational` is rewritten as `relate`.
Step 4: Fine-tuning by the following rules:
For the words ending with `e`, if the number of consonants is greater than 1 except for the first letter and the last letter, the last letter `e` is removed, for example, `relate` is changed to `relat`.

Fig. 2.5 The Porter stemming algorithm

2.3 Data Annotation

Data annotation is the foundation of supervised machine learning methods. In general, if the scale of annotated data is larger, the quality is higher, and if the coverage is broader, the performance of the trained model will be better. For different text mining tasks, the standards and specifications for data annotation are different, as is the complexity. For example, only category labels need to be annotated on each document for text classification tasks, while for some complex tasks, much more information needs to be marked, e.g., the boundary and type of each "entity" in the records should be marked for the analysis of electronic medical records. The "entity" mentioned here is not just the named entity (person

name, place name, organization name, time, number, etc.), as there are also many specialized terms in the medical field, such as disease names, the presence of certain symptoms, the absence certain symptoms, the frequency with which some symptoms occur, the factors of deterioration, irrelevant factors, and the degree. See the following two examples:[6]

(1) *Mr. Shinaberry is a 73-year-old gentleman who returned to* $[\text{Surgluthe Leon Calcner Healthcare}]_{Hosp}$ *to the emergency room on* $[\text{9/9/02}]_{Time}$ *with* $[\text{crescendo spontaneous angina}]_{Sym}$ *and* $[\text{shortness of breath}]_{Sym}$*. He is* $[\text{three-and-one-half months}]_{Dur}$ *after a presentation with* $[\text{subacute left circumflex thrombosis}]_{Dis}$*,*$[\text{ischemic mitral regurgitation}]_{Dis}$*,* $[\text{pulmonary edema}]_{Dis}$ *and a small* $[\text{nontransmural myocardial infarction}]_{Dis}$*. [Dilatation of the left circumflex*$]_{Treat}$ *resulted in extensive dissection but with eventual achievement of a very good* $[\text{angiographic and clinical result}]_{TR}$ *after* $[\text{placement of multiple stents}]_{Treat}$*, and his course was that of gradual recovery and uneventful return home.*

(2) *Mr. Brunckhorst is a 70-year-old man who recently had been experiencing an increase in frequency of* $[\text{chest pain}]_{Sym}$ *with exertion. He was administered an* $[\text{exercise tolerance test}]_{Test}$ *that was predictive of* $[\text{ischemia}]_{TR}$ *and revealed an* $[\text{ejection fraction of approximately 50\%}]_{TR}$*. As a result of his* $[\text{positive exercise tolerance test}]_{TR}$*, he was referred for* $[\text{cardiac catheterization}]_{Treat}$ *on March 1998, which revealed three* $[\text{vessel coronary artery disease}]_{Dis}$*. At this time, he was referred to the* $[\text{cardiac surgery service}]_{Treat}$ *for revascularization.*

In the examples, the label $Time$ indicates time, Sym indicates the presence of such symptoms, $Hosp$ indicates the name of the hospital, $Test$ indicates laboratory tests, TR indicates the results of the laboratory tests, Dis indicates the name of the disease, $Treat$ indicates the method of treatment, and Dur indicates the duration.

In the task of analyzing electronic medical records, usually, more than 20 different labels are defined. When annotating, it is often necessary to develop an annotation tool that can not only annotate the boundaries and types of all "entities" but also the relationships between them. In example (1) above, an annotation tool gives the relation graph shown in Fig. 2.6.

Of course, this type of relation graph is convenient and intuitive for annotators and domain experts to check and annotate. In fact, all specific marks are stored in the system. It is difficult to complete this kind of annotation task, which requires guidance based on professional knowledge, without the involvement of experts in the field.

For research on multimodal automatic summarization methods, we annotated a dataset including text, images, audio, and video. Different from synchronous multimodal data (e.g., movies), the dataset consists of asynchronous multimodal data, i.e., the pictures and sentences in the text or video and the sentences do not have

[6] https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/.

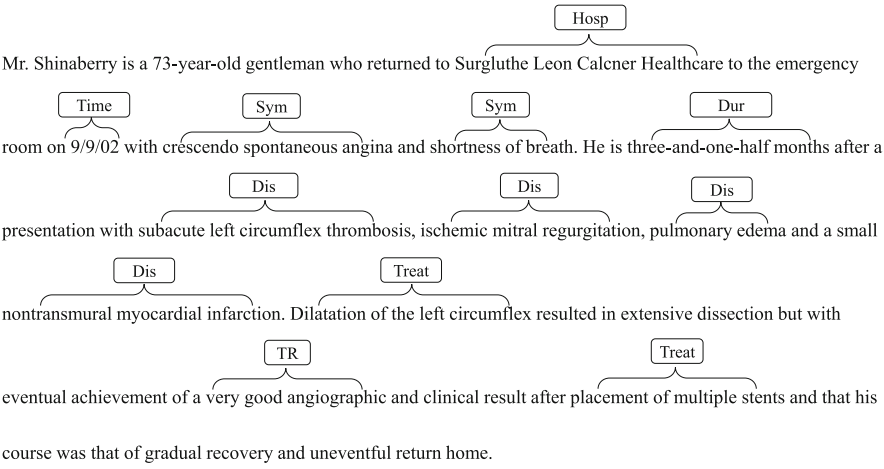

Fig. 2.6 An example of medical record annotation

a one-to-one correspondence. The dataset is centered on topics from Chinese and English news. There are multiple news documents with pictures for the same topic. For each topic, word-limited Chinese and English text summaries are presented.

During the data collection process, we chose 25 news topics in Chinese and English over the past 5 years, such as the Ebola virus in Africa, protesting against the deployment of the "Sade" antimissile system, or Li Na winning the Australian Open Tennis Championship. For each topic, we collected 20 news documents and 5 to 10 videos for the same period, making sure that the collected news texts were not significantly different in length. Generally, the length of each news item did not exceed 1,000 Chinese characters (or English words), and each video was within 2 min in length. The main reason for these restrictions is that overly long text or video will seriously increase the difficulty of manual annotation, which may lead to a too great divergence in the results annotated by different people.

During the annotation, the annotation policies given by the Document Understanding Conference and Text Analysis Conference were used for reference, and ten graduate students were invited to annotate the corpus. They were asked to read the news documents first, watch the videos on the same topic, and then write a summary independently. The policies for writing the summary are as follows: (1) the summary should retain the most important information from the news documents and videos; (2) there should be little to no redundant information in the summary; (3) the summary should have good readability; and (4) the summary should be within the length limitation (the Chinese summary does not exceed 500 Chinese characters, and the English abstract does not exceed 300 English words).

In the end, for each topic, three summaries independently written by different annotators were selected as reference answers.

At present, most of summaries generated by the existing automatic summarization systems are text without any other modal information, such as images.

Considering that a multimodal summary can enhance the user experience, we have also presented the summary data by text and picture. The annotation of this dataset involves two tasks: the writing of text summaries and the selection of pictures. The requirements for text summaries are the same as the methods described previously. To select the picture, two graduate students were invited to independently pick out the three most important pictures for each topic, and then we asked the third annotator to select three pictures as the final reference based on the results from the first two annotators. The basic policies for selecting pictures are that the pictures should be closely related to the news topic and they should be closely related to the content of the text summary.

The abovementioned corpora for summarization studies have been released on the following website: http://www.nlpr.ia.ac.cn/cip/dataset.htm/. Readers who are interested in multimodal summarization can download it from this website.

In summary, data annotation is a time-consuming and laborious task that often requires considerable manpower and financial support, so data sharing is particularly important. The methods introduced in this section are just examples, and more detailed specifications, standards, and instructions are required in data annotation. For many complex annotation tasks, developing convenient and easy-to-use annotation tools is a basic requirement for annotating large-scale data.

2.4 Basic Tools of NLP

As mentioned earlier, text mining involves many techniques from NLP, pattern classification, and machine learning and is one of technology with a clear application goal in across domains. Regardless of technologies applied for data preprocessing and annotation as described earlier or for the realization of data mining methods as will be described later, many basic techniques and tools are required, such as word segmenters, syntactic parsers, part-of-speech taggers, and chunkers. Some NLP methods are briefly introduced in the following.

2.4.1 Tokenization and POS Tagging

The purpose of tokenization is to separate text into a sequence of "words," which are usually called "tokens." The tokens include a string of successive alphanumeric characters, numbers, hyphens, and apostrophes. For example, "that's" will be separated into two tokens, that, 's; "rule-based" will be divided into three tokens, rule, -, based; and "TL-WR700N" will be divided into three tokens, TL, -, WR700N. The NLTK toolkit provides a tokenization package.[7]

[7] https://www.nltk.org/api/nltk.tokenize.html.

As we know, a word is usually expressed in different forms in the documents because of grammatical reasons, such as take, takes, taken, took, and taking. And also, many words can derive different expressions with the same meaning, such as token, tokenize, and tokenization. In practice, especially when using statistical methods, it is often necessary to detect the words sharing the same stem and meaning but in different forms and consider them the same when performing semantic understanding tasks. So, stemming and lemmatization are usually necessitated to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.[8]

For many Asian languages, such as Chinese, Japanese, Korean, and Vietnamese, the tokenization is usually expressed as word segmentation because their words are not separated by white spaces. The following takes the Chinese as example to briefly introduce the methods to segment Chinese words.

The Chinese word segmentation (CWS) is usually the first step in Chinese text processing, as noted above. There has been much research on CWS methods. From the early dictionary-based segmentation methods (such as the maximum matching method and shortest path segmentation method), to the statistical segmentation method based on *n*-gram, to the character-based CWS method later, dozens of segmentation methods have been proposed. Among them, the character-based CWS method is a landmark and an innovative method. The basic idea is that there are only four possible positions for any unit in a sentence, including Chinese characters, punctuation, digits, and any letters (collectively referred to as a "character"): the first position of the word (marked as B), the last position of the word (marked as E), the middle position of the word (marked as M), or a single character word (marked as S). B, E, M, and S are thus called the word position symbols. B and E always appear in pairs. Please see the following examples:

Chinese sentence: 约翰在北京见到了玛丽。(John met Mary in Beijing)
Segmentation result: 约 翰(John)/ 在(in)/ 北 京(Beijing)/ 见 到 了(met)/ 玛 丽(Mary)。
The segmentation results can be represented by position symbols: 约/B 翰/E 在/S 北/B 京/E 见/B 到/M 了/E 玛/B 丽/E 。/S

In this way, the task of CWS becomes the task of sequence labeling, and the classifier can be trained with large-scale labeled samples to carry out the task of labeling every unit in the text as a unique word position symbol. In practice, people also try to fuse or integrate several methods, such as the combination of the *n*-gram-based generative method and the character-based discriminative method (Wang et al. 2012) or the combination of the character-based method and the deep learning method, to establish a word segmenter with better performance.

Part-of-speech tagging refers to automatically tagging each word in a sentence with a part-of-speech category. For example, the sentence "天空是蔚蓝的(the sky is blue.)" is annotated as "天空/NN 是/NV 蔚蓝/AA 的/Au x 。/PU" after word

[8] https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html.

segmentation and part-of-speech tagging. The symbol NN represents noun, VV represents verb, AA represents adjective, Aux represents structural auxiliary, and PU represents punctuation. Part-of-speech tagging is the premise and foundation of syntactic parsing; it is also an important feature of text representation and is of great help to named entity recognition, relation extraction, and text sentiment analysis.

Part-of-speech tagging is a typical problem of sequence tagging. For Chinese text, this task is closely related to automatic word segmentation. Therefore, these two tasks are integrated in many CWS toolkits and are even achieved by a single model, such as the early CWS method based on the hidden Markov model (HMM).

At present, some CWS and part-of-speech tagging tools can be found in the following websites:

https://github.com/FudanNLP/fnlp
http://www.nlpr.ia.ac.cn/cip/software.htm
https://nlp.stanford.edu/software/tagger.shtml

In recent years, in deep learning methods or neural network-based methods, the text can be dealt with in character level or in sub-word level, but not in word level. So the tokenization and POS tagging can also be skipped.

2.4.2 Syntactic Parser

Syntactic parsing includes the tasks of constituent, or phrase structure, parsing, and dependency parsing. The purpose of phrase structure parsing is to automatically analyze the phrase structure relation in a sentence and to output the syntactic structure tree of the parsing sentence. The purpose of dependency parsing is to automatically analyze the relation of semantic dependency between words in a sentence. For example, Fig. 2.7 is a phrase structure tree of the sentence "The policemen have arrived at the scene and are carefully investigating the cause of the accident." The node symbols VV, NN, ADVP, NP, VP, and PU in Fig. 2.7 are part-of-speech symbols and phrase markers, respectively. IP is the root node symbol of the sentence. Figure 2.8 is the dependency tree corresponding to this sentence.

The arrow in Fig. 2.8 indicates the dependency (or domination) relation. The starting end of the arrow is the dominant word, and the pointing end of the arrow is the dominant word. The symbols on the directed arcs indicate the type of dependency relation. SBJ indicates a subject relation, i.e., the word at the end of the arrow is the subject of the word at the start of the arrow. OBJ indicates the object relation, that is, the word at the end of the arrow is the object of the word at the start of the arrow. VMOD indicates the verb modification relation, that is, the word at the end of the arrow modifies the verb at the beginning of the arrow. NMOD is a noun modification relation, that is, the word at the end of the arrow modifies the noun at the beginning of the arrow. ROOT denotes the root node of the clause. PU denotes the punctuation mark of the clause.

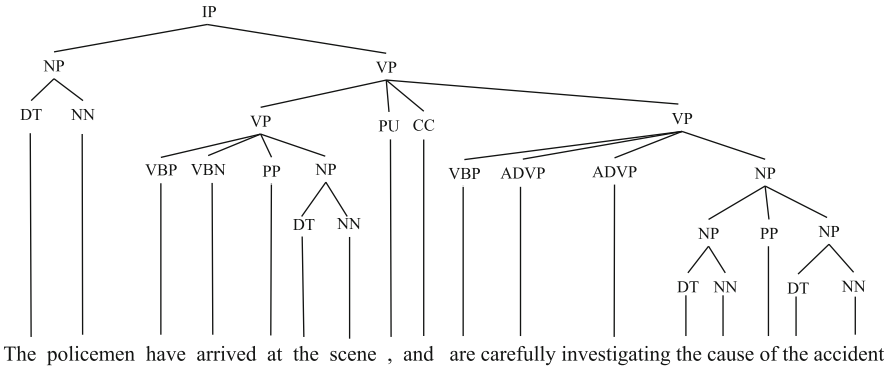

Fig. 2.7 An example of a phrase structure parsing tree

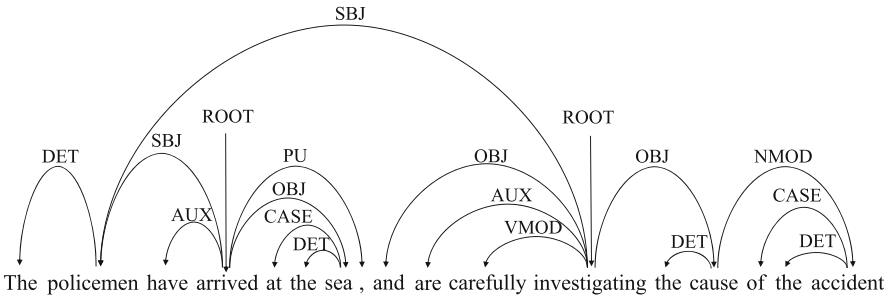

Fig. 2.8 An example of a dependency parsing tree

The phrase structure tree of a sentence can be converted into a dependency tree one by one. The basic idea of the conversion can be described as follows: first, determine the core predicate of the sentence as the only root node of the sentence, and then define the extraction rule of the central word. Next, the central word of each phrase is extracted, and the noncenter word is dominated by the central word.

In NLP, the phrase structure analyzer is usually called the syntactic parser, and the dependency analyzer is called the dependency parser.

The following web pages provide some parsers:

Berkeley Parser: https://github.com/nikitakit/self-attentive-parser
Charniak Parser: https://www.cs.brown.edu/people/ec/#software
http://www.nlpr.ia.ac.cn/cip/software.htm

The syntactic parser is usually employed to parse a complete sentence, and ultimately, we hope to obtain a full parsing tree for the sentence, so it is also called full parsing. In practice, sometimes it is not necessary to obtain a complete syntactic parsing tree but only to identify the basic noun phrase (base NP) or basic verb phrase (base VP) included in the sentence. For example, the sentence *Foreign-funded enterprises also play an important role in China's economy* contains the base

NPs *foreign-funded enterprises, China's economy*, and *important role* and contains the base VP *play*. The parsing technique for identifying a specific type of phrase in a sentence is usually called shallow parsing. At present, the shallow parsing method in use is more similar to the character-based word segmentation method. The tagging unit can be either the word or the character. The word or character position tag can adopt four tagging systems using B, E, M, and S and can also adopt three tagging systems using B, I, and O. For example, NP-B denotes the first word (character) of a base NP, NP-I denotes that the word (character) is inside the base NP, and NP-O denotes that the word (character) does not belong to the NP. The classifier model is similar to the methods used in CWS and named entity recognition. Readers can refer to Chap. 9 for a detailed introduction of named entity recognition methods.

2.4.3 N-gram Language Model

N-gram is a traditional language model (LM) that plays a very important role in NLP. The basic idea is as follows: for a character string (phrase, sentence, or fragment) $s = w_1 w_2 \cdots w_l$ composed of l (l is a natural number, $l \geq 2$) basic statistical units, its probability can be calculated by the following formula:

$$\begin{aligned} p(s) &= p(w_1)p(w_2|w_1)p(w_3|w_1 w_2)\cdots p(w_l|w_1 \cdots w_{l-1}) \\ &= \prod_{i=1}^{l} p(w_i|w_1 \cdots w_{i-1}) \end{aligned} \tag{2.1}$$

The *basic statistical units* mentioned here may be characters, words, punctuation, digits, or any other symbols constituting a sentence, or even phrases, part-of-speech tags, etc., which are collectively referred to "words" for convenience of expression. In Eq. (2.1), it means that the probability of generating the i-th ($1 \leq i \leq l$) word is determined by the previously (the "previously" usually refers to the left in the written order of the words) generated $i-1$ words $w_1 w_2 \cdots w_{i-1}$. With increasing sentence length, the historical number of conditional probabilities increases exponentially. To simplify the complexity of the calculation, it is assumed that the probability of the current word is only related to the previous $n-1$ (n is an integer, $1 \leq n \leq l$) words. Thus, Eq. (2.1) becomes

$$p(s) = \prod_{i-1}^{l} p(w_i|w_1 \cdots w_{i-1}) \approx \prod_{i=1}^{l} p(w_i|w_{i-1}) \tag{2.2}$$

When $n = 1$, the probability of word w_i appearing at the i-th position is independent of the previous words, and the sentence is a sequence of independent words. This calculation model is usually called a one-gram model, which is recorded as a unigram, unigram, or monogram. Each word is a unigram. When $n = 2$, the probability of word w_i appearing at the i-th position is only related to the previous

word w_{i-1}. This calculation model is called the two-gram model. Two adjacent co-occurrence words are called two-grams, usually signed as bigrams or bi-grams. For example, for the sentence *We helped her yesterday*, the following sequence of words, *We helped*, *helped her*, and *her yesterday*, are all bigrams. In this case, the sentence is regarded as a chain composed of bigrams, called a first-order Markov chain. By that analogy, when $n = 3$, the probability of the word w_i appearing at the i-th position is only related to the previous word w_{i-1} and word w_{i-2} ($i \geq 2$). This calculation model is called a three-gram model. The sequence of three adjacent co-occurrence words is called three grams, usually signed as trigrams or tri-grams. Sequences composed of trigrams can be regarded as second-order Markov chains.

When calculating the n-gram model, a key problem is smoothing the data to avoid the problems caused by zero probability events (n-gram). For this reason, researchers have proposed several data smoothing methods, such as additive smoothing, discounting methods, and deleted interpolation methods. At the same time, to eliminate the negative influence of training samples from different fields, topics, and types on the model's performance, researchers have also proposed methods for language model adaptation, which will not be described in detail here. Readers can refer to (Chen and Goodman 1999) and (Zong 2013) if interested.

The neural network language model (NNLM) has played an important role in NLP in recent years. For details about this model, please refer to Chap. 3 in this book.

2.5 Further Reading

In addition to the NLP technologies introduced above, word sense disambiguation (WSD), semantic role labeling (SRL), and text entailment are also helpful for text data mining, but their performance has not reached a high level (e.g., the accuracy of semantic role labeling for normal text is only 80%). Relevant technical methods have been described in many NLP publications. The readers can refer to (Manning and Schütze 1999; Jurafsky and Martin 2008; Zong 2013) if necessary.

Exercises

2.1 Please collect some articles from newspapers and some text from social network sites, such as Twitter, QQ, or other microblog sites. Compare the different language expressions and summarize your observations.

2.2 Collect some sentences, parse them using a constituent parser and a dependency parser, and then compare the different results, i.e., the syntactic parsing tree and the dependency of words in the sentences.

2.3 Collect some text from social network sites in which there are some noise and stop words. Please make a list of the stop words and create a standard or policy to determine what words are noise. Then, implement a program to remove the stop words and noise words from the given text. Aim to deliver higher generalization for the algorithm of the program.

2.4 Collect some corpora, tokenize it by implementing an existing tool or a program, and extract all n-grams in the corpora ($n \in N$ and $1 < n \leq 4$).

2.5 Collect some medical instructions or some chapters from text books for medical students and annotate all named entities, other terms, and their relations in the collected instructions or text corpora.

2.6 Use a Chinese word segmenter to perform Chinese word segmentation for different styles of Chinese corpora, such as from public newspapers, specific technical domains, and social network sites. Analyze the segmentation results, and evaluate the correct rate of the Chinese word segmenter. How do the results compare to the segmentation results for the same corpora when using a different segmenter?