Chapter 8

Applications of sequence labeling

Sequence labeling has applications throughout natural language processing. This chapter focuses on part-of-speech tagging, morpho-syntactic attribute tagging, named entity recognition, and tokenization. It also touches briefly on two applications to interactive settings: dialogue act recognition and the detection of code-switching points between languages.

8.1 Part-of-speech tagging

The **syntax** of a language is the set of principles under which sequences of words are judged to be grammatically acceptable by fluent speakers. One of the most basic syntactic concepts is the **part-of-speech** (POS), which refers to the syntactic role of each word in a sentence. This concept was used informally in the previous chapter, and you may have some intuitions from your own study of English. For example, in the sentence *We like vegetarian sandwiches*, you may already know that *we* and *sandwiches* are nouns, *like* is a verb, and *vegetarian* is an adjective. These labels depend on the context in which the word appears: in *she eats like a vegetarian*, the word *like* is a preposition, and the word *vegetarian* is a noun.

Parts-of-speech can help to disentangle or explain various linguistic problems. Recall Chomsky's proposed distinction in chapter 6:

(8.1) a. Colorless green ideas sleep furiously.
b. * Ideas colorless furiously green sleep.

One difference between these two examples is that the first contains part-of-speech transitions that are typical in English: adjective to adjective, adjective to noun, noun to verb, and verb to adverb. The second example contains transitions that are unusual: noun to adjective and adjective to verb. The ambiguity in a headline like,

(8.2) Teacher Strikes Idle Children

can also be explained in terms of parts of speech: in the interpretation that was likely intended, *strikes* is a noun and *idle* is a verb; in the alternative explanation, *strikes* is a verb and *idle* is an adjective.

Part-of-speech tagging is often taken as a early step in a natural language processing pipeline. Indeed, parts-of-speech provide features that can be useful for many of the tasks that we will encounter later, such as parsing (chapter 10), coreference resolution (chapter 15), and relation extraction (chapter 17).

8.1.1 Parts-of-Speech

The **Universal Dependencies** project (UD) is an effort to create syntactically-annotated corpora across many languages, using a single annotation standard (Nivre et al., 2016). As part of this effort, they have designed a part-of-speech **tagset**, which is meant to capture word classes across as many languages as possible.[1] This section describes that inventory, giving rough definitions for each of tags, along with supporting examples.

Part-of-speech tags are **morphosyntactic**, rather than semantic, categories. This means that they describe words in terms of how they pattern together and how they are internally constructed (e.g., what suffixes and prefixes they include). For example, you may think of a noun as referring to objects or concepts, and verbs as referring to actions or events. But events can also be nouns:

(8.3) . . . the **howling** of the **shrieking** storm.

Here *howling* and *shrieking* are events, but grammatically they act as a noun and adjective respectively.

The Universal Dependency part-of-speech tagset

The UD tagset is broken up into three groups: open class tags, closed class tags, and "others."

Open class tags Nearly all languages contain nouns, verbs, adjectives, and adverbs.[2] These are all **open word classes**, because new words can easily be added to them. The UD tagset includes two other tags that are open classes: proper nouns and interjections.

- **Nouns** (UD tag: NOUN) tend to describe entities and concepts, e.g.,

[1] The UD tagset builds on earlier work from Petrov et al. (2012), in which a set of twelve universal tags was identified by creating mappings from tagsets for individual languages.

[2] One prominent exception is Korean, which some linguists argue does not have adjectives Kim (2002).

(8.4) **Toes** are scarce among veteran **blubber men**.

In English, nouns tend to follow determiners and adjectives, and can play the subject role in the sentence. They can be marked for the plural number by an *-s* suffix.

- **Proper nouns** (PROPN) are tokens in names, which uniquely specify a given entity,

  (8.5) "**Moby Dick**?" shouted **Ahab**.

- **Verbs** (VERB), according to the UD guidelines, "typically signal events and actions." But they are also defined grammatically: they "can constitute a minimal predicate in a clause, and govern the number and types of other constituents which may occur in a clause."[3]

  (8.6) "Moby Dick?" **shouted** Ahab.

  (8.7) Shall we **keep chasing** this murderous fish?

  English verbs tend to come in between the subject and some number of direct objects, depending on the verb. They can be marked for **tense** and **aspect** using suffixes such as *-ed* and *-ing*. (These suffixes are an example of **inflectional morphology**, which is discussed in more detail in § 9.1.4.)

- **Adjectives** (ADJ) describe properties of entities,

  (8.8) a. Shall we keep chasing this **murderous** fish?
  b. Toes are **scarce** among **veteran** blubber men.

  In the second example, *scarce* is a predicative adjective, linked to the subject by the **copula verb** *are*. In contrast, *murderous* and *veteran* are attributive adjectives, modifying the noun phrase in which they are embedded.

- **Adverbs** (ADV) describe properties of events, and may also modify adjectives or other adverbs:

  (8.9) a. It is not down on any map; true places **never** are.
  b. ... **treacherously** hidden beneath the loveliest tints of azure
  c. Not drowned **entirely**, though.

- **Interjections** (INTJ) are used in exclamations, e.g.,

  (8.10) **Aye aye**! it was that accursed white whale that razed me.

[3] http://universaldependencies.org/u/pos/VERB.html

Closed class tags Closed word classes rarely receive new members. They are sometimes referred to as **function words** — as opposed to **content words** — as they have little lexical meaning of their own, but rather, help to organize the components of the sentence.

- **Adpositions** (ADP) describe the relationship between a complement (usually a noun phrase) and another unit in the sentence, typically a noun or verb phrase.

  (8.11) a. Toes are scarce **among** veteran blubber men.
  b. It is not **down on** any map.
  c. Give not thyself **up** then.

  As the examples show, English generally uses prepositions, which are adpositions that appear before their complement. (An exception is *ago*, as in, *we met three days **ago***). Postpositions are used in other languages, such as Japanese and Turkish.
- **Auxiliary verbs** (AUX) are a closed class of verbs that add information such as tense, aspect, person, and number.

  (8.12) a. **Shall** we keep chasing this murderous fish?
  b. What the white whale was to Ahab, **has been** hinted.
  c. Ahab **must** use tools.
  d. Meditation and water **are** wedded forever.
  e. Toes **are** scarce among veteran blubber men.

  The final example is a copula verb, which is also tagged as an auxiliary in the UD corpus.
- **Coordinating conjunctions** (CCONJ) express relationships between two words or phrases, which play a parallel role:

  (8.13) Meditation **and** water are wedded forever.

- **Subordinating conjunctions** (SCONJ) link two clauses, making one syntactically subordinate to the other:

  (8.14) It is the easiest thing in the world for a man to look as **if** he had a great secret in him.

  Note that
- **Pronouns** (PRON) are words that substitute for nouns or noun phrases.

  (8.15) a. Be **it what it** will, **I**'ll go to **it** laughing.

b. **I** try all things, **I** achieve **what I** can.

The example includes the personal pronouns *I* and *it*, as well as the relative pronoun *what*. Other pronouns include *myself*, *somebody*, and *nothing*.

- **Determiners** (DET) provide additional information about the nouns or noun phrases that they modify:

(8.16) a. What **the** white whale was to Ahab, has been hinted.
b. It is not down on **any** map.
c. I try **all** things ...
d. Shall we keep chasing **this** murderous fish?

Determiners include articles (*the*), possessive determiners (*their*), demonstratives (*this murderous fish*), and quantifiers (*any map*).

- **Numerals** (NUM) are an infinite but closed class, which includes integers, fractions, and decimals, regardless of whether spelled out or written in numerical form.

(8.17) a. How then can this **one** small heart beat.
b. I am going to put him down for the **three hundredth**.

- **Particles** (PART) are a catch-all of function words that combine with other words or phrases, but do not meet the conditions of the other tags. In English, this includes the infinitival *to*, the possessive marker, and negation.

(8.18) a. Better **to** sleep with a sober cannibal than a drunk Christian.
b. So man**'s** insanity is heaven**'s** sense
c. It is **not** down on any map

As the second example shows, the possessive marker is not considered part of the same token as the word that it modifies, so that *man's* is split into two tokens. (Tokenization is described in more detail in § 8.4.) A non-English example of a particle is the Japanese question marker *ka*:[4]

(8.19) *Sensei desu ka*
Teacher is ?
Is she a teacher?

[4]In this notation, the first line is the transliterated Japanese text, the second line is a token-to-token **gloss**, and the third line is the translation.

Other The remaining UD tags include punctuation (PUN) and symbols (SYM). Punctuation is purely structural — e.g., commas, periods, colons — while symbols can carry content of their own. Examples of symbols include dollar and percentage symbols, mathematical operators, emoticons, emojis, and internet addresses. A final catch-all tag is X, which is used for words that cannot be assigned another part-of-speech category. The X tag is also used in cases of **code switching** (between languages), described in § 8.5.

Other tagsets

Prior to the Universal Dependency treebank, part-of-speech tagging was performed using language-specific tagsets. The dominant tagset for English was designed as part of the **Penn Treebank** (PTB), and it includes 45 tags — more than three times as many as the UD tagset. This granularity is reflected in distinctions between singular and plural nouns, verb tenses and aspects, possessive and non-possessive pronouns, comparative and superlative adjectives and adverbs (e.g., *faster*, *fastest*), and so on. The Brown corpus includes a tagset that is even more detailed, with 87 tags (Francis, 1964), including special tags for individual auxiliary verbs such as *be*, *do*, and *have*.

Different languages make different distinctions, and so the PTB and Brown tagsets are not appropriate for a language such as Chinese, which does not mark the verb tense (Xia, 2000); nor for Spanish, which marks every combination of person and number in the verb ending; nor for German, which marks the case of each noun phrase. Each of these languages requires more detail than English in some areas of the tagset, and less in other areas. The strategy of the Universal Dependencies corpus is to design a coarse-grained tagset to be used across all languages, and then to additionally annotate language-specific **morphosyntactic attributes**, such as number, tense, and case. The attribute tagging task is described in more detail in § 8.2.

Social media such as Twitter have been shown to require tagsets of their own (Gimpel et al., 2011). Such corpora contain some tokens that are not equivalent to anything encountered in a typical written corpus: e.g., emoticons, URLs, and hashtags. Social media also includes dialectal words like *gonna* ('going to', e.g. *We gonna be fine*) and *Ima* ('I'm going to', e.g., *Ima tell you one more time*), which can be analyzed either as non-standard orthography (making tokenization impossible), or as lexical items in their own right. In either case, it is clear that existing tags like NOUN and VERB cannot handle cases like *Ima*, which combine aspects of the noun and verb. Gimpel et al. (2011) therefore propose a new set of tags to deal with these cases.

8.1.2 Accurate part-of-speech tagging

Part-of-speech tagging is the problem of selecting the correct tag for each word in a sentence. Success is typically measured by accuracy on an annotated test set, which is simply the fraction of tokens that were tagged correctly.

Baselines

A simple baseline for part-of-speech tagging is to choose the most common tag for each word. For example, in the Universal Dependencies treebank, the word *talk* appears 96 times, and 85 of those times it is labeled as a VERB: therefore, this baseline will always predict VERB for this word. For words that do not appear in the training corpus, the baseline simply guesses the most common tag overall, which is NOUN. In the Penn Treebank, this simple baseline obtains accuracy above 92%. A more rigorous evaluation is the accuracy on **out-of-vocabulary words**, which are not seen in the training data. Tagging these words correctly requires attention to the context and the word's internal structure.

Contemporary approaches

Conditional random fields and structured perceptron perform at or near the state-of-the-art for part-of-speech tagging in English. For example, (Collins, 2002) achieved 97.1% accuracy on the Penn Treebank, using a structured perceptron with the following base features (originally introduced by Ratnaparkhi (1996)):

- current word, w_m
- previous words, w_{m-1}, w_{m-2}
- next words, w_{m+1}, w_{m+2}
- previous tag, y_{m-1}
- previous two tags, (y_{m-1}, y_{m-2})
- for rare words:
  - first k characters, up to $k = 4$
  - last k characters, up to $k = 4$
  - whether w_m contains a number, uppercase character, or hyphen.

Similar results for the PTB data have been achieved using conditional random fields (CRFs; Toutanova et al., 2003).

More recent work has demonstrated the power of neural sequence models, such as the **long short-term memory (LSTM)** (§ 7.6). Plank et al. (2016) apply a CRF and a bidirectional LSTM to twenty-two languages in the UD corpus, achieving an average accuracy of 94.3% for the CRF, and 96.5% with the bi-LSTM. Their neural model employs three types of embeddings: fine-tuned word embeddings, which are updated during training; pre-trained word embeddings, which are never updated, but which help to tag out-of-vocabulary words; and character-based embeddings. The character-based embeddings are computed by running an LSTM on the individual characters in each word, thereby capturing common orthographic patterns such as prefixes, suffixes, and capitalization. Extensive evaluations show that these additional embeddings are crucial to their model's success.

word	PTB tag	UD tag	UD attributes
The	DT	DET	DEFINITE=DEF PRONTYPE=ART
German	JJ	ADJ	DEGREE=POS
Expressionist	NN	NOUN	NUMBER=SING
movement	NN	NOUN	NUMBER=SING
was	VBD	AUX	MOOD=IND NUMBER=SING PERSON=3 TENSE=PAST VERBFORM=FIN
destroyed	VBN	VERB	TENSE=PAST VERBFORM=PART VOICE=PASS
as	IN	ADP	
a	DT	DET	DEFINITE=IND PRONTYPE=ART
result	NN	NOUN	NUMBER=SING
.	.	PUNCT	

Figure 8.1: UD and PTB part-of-speech tags, and UD morphosyntactic attributes. Example selected from the UD 1.4 English corpus.

8.2 Morphosyntactic Attributes

There is considerably more to say about a word than whether it is a noun or a verb: in English, verbs are distinguish by features such tense and aspect, nouns by number, adjectives by degree, and so on. These features are language-specific: other languages distinguish other features, such as **case** (the role of the noun with respect to the action of the sentence, which is marked in languages such as Latin and German[5]) and **evidentiality** (the source of information for the speaker's statement, which is marked in languages such as Turkish). In the UD corpora, these attributes are annotated as feature-value pairs for each token.[6]

An example is shown in Figure 8.1. The determiner *the* is marked with two attributes: PRONTYPE=ART, which indicates that it is an **article** (as opposed to another type of deter-

[5]Case is marked in English for some personal pronouns, e.g., ***She*** *saw* ***her***, *They* saw *them*.

[6]The annotation and tagging of morphosyntactic attributes can be traced back to earlier work on Turkish (Oflazer and Kuruöz, 1994) and Czech (Hajič and Hladká, 1998). MULTEXT-East was an early multilingual corpus to include morphosyntactic attributes (Dimitrova et al., 1998).

miner or pronominal modifier), and DEFINITE=DEF, which indicates that it is a **definite article** (referring to a specific, known entity). The verbs are each marked with several attributes. The auxiliary verb *was* is third-person, singular, past tense, finite (conjugated), and indicative (describing an event that has happened or is currently happenings); the main verb *destroyed* is in participle form (so there is no additional person and number information), past tense, and passive voice. Some, but not all, of these distinctions are reflected in the PTB tags VBD (past-tense verb) and VBN (past participle).

While there are thousands of papers on part-of-speech tagging, there is comparatively little work on automatically labeling morphosyntactic attributes. Faruqui et al. (2016) train a support vector machine classification model, using a minimal feature set that includes the word itself, its prefixes and suffixes, and type-level information listing all possible morphosyntactic attributes for each word and its neighbors. Mueller et al. (2013) use a conditional random field (CRF), in which the tag space consists of all observed combinations of morphosyntactic attributes (e.g., the tag would be DEF+ART for the word *the* in Figure 8.1). This massive tag space is managed by decomposing the feature space over individual attributes, and pruning paths through the trellis. More recent work has employed bidirectional LSTM sequence models. For example, Pinter et al. (2017) train a bidirectional LSTM sequence model. The input layer and hidden vectors in the LSTM are shared across attributes, but each attribute has its own output layer, culminating in a softmax over all attribute values, e.g. $y_t^{\text{NUMBER}} \in \{\text{SING}, \text{PLURAL}, \ldots\}$. They find that character-level information is crucial, especially when the amount of labeled data is limited.

Evaluation is performed by first computing recall and precision for each attribute. These scores can then be averaged at either the type or token level to obtain micro- or macro-F-MEASURE. Pinter et al. (2017) evaluate on 23 languages in the UD treebank, reporting a median micro-F-MEASURE of 0.95. Performance is strongly correlated with the size of the labeled dataset for each language, with a few outliers: for example, Chinese is particularly difficult, because although the dataset is relatively large (10^5 tokens in the UD 1.4 corpus), only 6% of tokens have any attributes, offering few useful labeled instances.

8.3 Named Entity Recognition

A classical problem in information extraction is to recognize and extract mentions of **named entities** in text. In news documents, the core entity types are people, locations, and organizations; more recently, the task has been extended to include amounts of money, percentages, dates, and times. In item 8.20a (Figure 8.2), the named entities include: *The U.S. Army*, an organization; *Atlanta*, a location; and *May 14, 1864*, a date. Named entity recognition is also a key task in **biomedical natural language processing**, with entity types including proteins, DNA, RNA, and cell lines (e.g., Collier et al., 2000; Ohta et al., 2002). Figure 8.2 shows an example from the GENIA corpus of biomedical research ab-

(8.20) a. *The U.S. Army captured Atlanta on May 14 , 1864*
B-ORG I-ORG I-ORG O B-LOC O B-DATE I-DATE I-DATE I-DATE

b. *Number of glucocorticoid receptors in lymphocytes and ...*
O O B-PROTEIN I-PROTEIN O B-CELLTYPE O ...

Figure 8.2: BIO notation for named entity recognition. Example (8.20b) is drawn from the GENIA corpus of biomedical documents (Ohta et al., 2002).

stracts.

A standard approach to tagging named entity spans is to use discriminative sequence labeling methods such as conditional random fields. However, the named entity recognition (NER) task would seem to be fundamentally different from sequence labeling tasks like part-of-speech tagging: rather than tagging each token, the goal in is to recover *spans* of tokens, such as *The United States Army*.

This is accomplished by the **BIO notation**, shown in Figure 8.2. Each token at the beginning of a name span is labeled with a B- prefix; each token within a name span is labeled with an I- prefix. These prefixes are followed by a tag for the entity type, e.g. B-LOC for the beginning of a location, and I-PROTEIN for the inside of a protein name. Tokens that are not parts of name spans are labeled as O. From this representation, the entity name spans can be recovered unambiguously. This tagging scheme is also advantageous for learning: tokens at the beginning of name spans may have different properties than tokens within the name, and the learner can exploit this. This insight can be taken even further, with special labels for the **l**ast tokens of a name span, and for **u**nique tokens in name spans, such as *Atlanta* in the example in Figure 8.2. This is called BILOU notation, and it can yield improvements in supervised named entity recognition (Ratinov and Roth, 2009).

Feature-based sequence labeling Named entity recognition was one of the first applications of conditional random fields (McCallum and Li, 2003). The use of Viterbi decoding restricts the feature function $\boldsymbol{f}(\boldsymbol{w}, \boldsymbol{y})$ to be a sum of local features, $\sum_m \boldsymbol{f}(\boldsymbol{w}, y_m, y_{m-1}, m)$, so that each feature can consider only local adjacent tags. Typical features include tag transitions, word features for w_m and its neighbors, character-level features for prefixes and suffixes, and "word shape" features for capitalization and other orthographic properties. As an example, base features for the word *Army* in the example in (8.20a) include:

$$(\text{CURR-WORD:}\textit{Army}, \text{PREV-WORD:}\textit{U.S.}, \text{NEXT-WORD:}\textit{captured}, \text{PREFIX-1:}\textit{A-},$$
$$\text{PREFIX-2:}\textit{Ar-}, \text{SUFFIX-1:}\textit{-y}, \text{SUFFIX-2:}\textit{-my}, \text{SHAPE:}\textit{Xxxx})$$

Features can also be obtained from a **gazetteer**, which is a list of known entity names. For example, the U.S. Social Security Administration provides a list of tens of thousands of

(1) 日文 章魚 怎麼 說?
Japanese octopus how say
How to say octopus in Japanese?

(2) 日 文章 魚 怎麼 說?
Japan essay fish how say

Figure 8.3: An example of tokenization ambiguity in Chinese (Sproat et al., 1996)

given names — more than could be observed in any annotated corpus. Tokens or spans that match an entry in a gazetteer can receive special features; this provides a way to incorporate hand-crafted resources such as name lists in a learning-driven framework.

Neural sequence labeling for NER Current research has emphasized neural sequence labeling, using similar LSTM models to those employed in part-of-speech tagging (Hammerton, 2003; Huang et al., 2015; Lample et al., 2016). The bidirectional LSTM-CRF (Figure 7.4 in § 7.6) does particularly well on this task, due to its ability to model tag-to-tag dependencies. However, Strubell et al. (2017) show that **convolutional neural networks** can be equally accurate, with significant improvement in speed due to the efficiency of implementing ConvNets on **graphics processing units (GPUs)**. The key innovation in this work was the use of **dilated convolution**, which is described in more detail in § 3.4.

8.4 Tokenization

A basic problem for text analysis, first discussed in § 4.3.1, is to break the text into a sequence of discrete tokens. For alphabetic languages such as English, deterministic scripts usually suffice to achieve accurate tokenization. However, in logographic writing systems such as Chinese script, words are typically composed of a small number of characters, without intervening whitespace. The tokenization must be determined by the reader, with the potential for occasional ambiguity, as shown in Figure 8.3. One approach is to match character sequences against a known dictionary (e.g., Sproat et al., 1996), using additional statistical information about word frequency. However, no dictionary is completely comprehensive, and dictionary-based approaches can struggle with such out-of-vocabulary words.

Chinese word segmentation has therefore been approached as a supervised sequence labeling problem. Xue et al. (2003) train a logistic regression classifier to make independent segmentation decisions while moving a sliding window across the document. A set of rules is then used to convert these individual classification decisions into an overall tokenization of the input. However, these individual decisions may be globally suboptimal, motivating a structure prediction approach. Peng et al. (2004) train a conditional random

field to predict labels of START or NONSTART on each character. More recent work has employed neural network architectures. For example, Chen et al. (2015) use an LSTM-CRF architecture, as described in § 7.6: they construct a trellis, in which each tag is scored according to the hidden state of an LSTM, and tag-tag transitions are scored according to learned transition weights. The best-scoring segmentation is then computed by the Viterbi algorithm.

8.5 Code switching

Multilingual speakers and writers do not restrict themselves to a single language. **Code switching** is the phenomenon of switching between languages in speech and text (Auer, 2013; Poplack, 1980). Written code switching has become more common in online social media, as in the following extract from the website of Canadian President Justin Trudeau:[7]

(8.21) *Although everything written on this site est disponible en anglais*
is available in English
and in French, my personal videos seront bilingues
will be bilingual

Accurately analyzing such texts requires first determining which languages are being used. Furthermore, quantitative analysis of code switching can provide insights on the languages themselves and their relative social positions.

Code switching can be viewed as a sequence labeling problem, where the goal is to label each token as a candidate switch point. In the example above, the words *est*, *and*, and *seront* would be labeled as switch points. Solorio and Liu (2008) detect English-Spanish switch points using a supervised classifier, with features that include the word, its part-of-speech in each language (according to a supervised part-of-speech tagger), and the probabilities of the word and part-of-speech in each language. Nguyen and Dogruöz (2013) apply a conditional random field to the problem of detecting code switching between Turkish and Dutch.

Code switching is a special case of the more general problem of word level language identification, which Barman et al. (2014) address in the context of trilingual code switching between Bengali, English, and Hindi. They further observe an even more challenging phenomenon: intra-word code switching, such as the use of English suffixes with Bengali roots. They therefore mark each token as either (1) belonging to one of the three languages; (2) a mix of multiple languages; (3) "universal" (e.g., symbols, numbers, emoticons); or (4) undefined.

[7] As quoted in `http://blogues.lapresse.ca/lagace/2008/09/08/justin-trudeau-really-parfait-bilingue/`, accessed August 21, 2017.

Speaker	Dialogue Act	Utterance
A	YES-NO-QUESTION	*So do you go college right now?*
A	ABANDONED	*Are yo-*
B	YES-ANSWER	*Yeah,*
B	STATEMENT	*It's my last year [laughter].*
A	DECLARATIVE-QUESTION	*You're a, so you're a senior now.*
B	YES-ANSWER	*Yeah,*
B	STATEMENT	*I'm working on my projects trying to graduate [laughter]*
A	APPRECIATION	*Oh, good for you.*
B	BACKCHANNEL	*Yeah.*

Figure 8.4: An example of dialogue act labeling (Stolcke et al., 2000)

8.6 Dialogue acts

The sequence labeling problems that we have discussed so far have been over sequences of word tokens or characters (in the case of tokenization). However, sequence labeling can also be performed over higher-level units, such as **utterances**. **Dialogue acts** are labels over utterances in a dialogue, corresponding roughly to the speaker's intention — the utterance's **illocutionary force** (Austin, 1962). For example, an utterance may state a proposition (*it is not down on any map*), pose a question (*shall we keep chasing this murderous fish?*), or provide a response (*aye aye!*). Stolcke et al. (2000) describe how a set of 42 dialogue acts were annotated for the 1,155 conversations in the Switchboard corpus (Godfrey et al., 1992).[8]

An example is shown in Figure 8.4. The annotation is performed over UTTERANCES, with the possibility of multiple utterances per **conversational turn** (in cases such as interruptions, an utterance may split over multiple turns). Some utterances are clauses (e.g., *So do you go to college right now?*), while others are single words (e.g., *yeah*). Stolcke et al. (2000) report that hidden Markov models (HMMs) achieve 96% accuracy on supervised utterance segmentation. The labels themselves reflect the conversational goals of the speaker: the utterance *yeah* functions as an answer in response to the question *you're a senior now*, but in the final line of the excerpt, it is a **backchannel** (demonstrating comprehension).

For task of dialogue act labeling, Stolcke et al. (2000) apply a hidden Markov model. The probability $\text{p}(\boldsymbol{w}_m \mid y_m)$ must generate the entire sequence of words in the utterance, and it is modeled as a trigram language model (§ 6.1). Stolcke et al. (2000) also account for acoustic features, which capture the **prosody** of each utterance — for example, tonal and rhythmic properties of speech, which can be used to distinguish dialogue acts such

[8]Dialogue act modeling is not restricted to speech; it is relevant in any interactive conversation. For example, Jeong et al. (2009) annotate a more limited set of **speech acts** in a corpus of emails and online forums.

as questions and answers. These features are handled with an additional emission distribution, $p(\boldsymbol{a}_m \mid y_m)$, which is modeled with a probabilistic decision tree (Murphy, 2012). While acoustic features yield small improvements overall, they play an important role in distinguish questions from statements, and agreements from backchannels.

Recurrent neural architectures for dialogue act labeling have been proposed by Kalchbrenner and Blunsom (2013) and Ji et al. (2016), with strong empirical results. Both models are recurrent at the utterance level, so that each complete utterance updates a hidden state. The recurrent-convolutional network of Kalchbrenner and Blunsom (2013) uses convolution to obtain a representation of each individual utterance, while Ji et al. (2016) use a second level of recurrence, over individual words. This enables their method to also function as a language model, giving probabilities over sequences of words in a document.

Exercises

1. Using the Universal Dependencies part-of-speech tags, annotate the following sentences. You may examine the UD tagging guidelines. Tokenization is shown with whitespace. Don't forget about punctuation.

   (8.22) a. I try all things , I achieve what I can .
   b. It was that accursed white whale that razed me .
   c. Better to sleep with a sober cannibal , than a drunk Christian .
   d. Be it what it will , I 'll go to it laughing .

2. Select three short sentences from a recent news article, and annotate them for UD part-of-speech tags. Ask a friend to annotate the same three sentences without looking at your annotations. Compute the rate of agreement, using the Kappa metric defined in § 4.5.2. Then work together to resolve any disagreements.

3. Choose one of the following morphosyntactic attributes: MOOD, TENSE, VOICE. Research the definition of this attribute on the universal dependencies website, `http://universaldependencies.org/u/feat/index.html`. Returning to the examples in the first exercise, annotate all verbs for your chosen attribute. It may be helpful to consult examples from an English-language universal dependencies corpus, available at `https://github.com/UniversalDependencies/UD_English-EWT/tree/master`.

4. Download a dataset annotated for universal dependencies, such as the English Treebank at `https://github.com/UniversalDependencies/UD_English-EWT/tree/master`. This corpus is already segmented into training, development, and test data.

a) First, train a logistic regression or SVM classifier using character suffixes: character n-grams up to length 4. Compute the recall, precision, and F-MEASURE on the development data.

b) Next, augment your classifier using the same character suffixes of the preceding and succeeding tokens. Again, evaluate your classifier on heldout data.

c) Optionally, train a Viterbi-based sequence labeling model, using a toolkit such as CRFSuite (`http://www.chokkan.org/software/crfsuite/`) or your own Viterbi implementation. This is more likely to be helpful for attributes in which agreement is required between adjacent words. For example, many Romance languages require gender and number agreement for determiners, nouns, and adjectives.

5. Provide BIO-style annotation of the named entities (person, place, organization, date, or product) in the following expressions:

(8.23) a. The third mate was Flask, a native of Tisbury, in Martha's Vineyard.

b. Its official Nintendo announced today that they Will release the Nintendo 3DS in north America march 27 (Ritter et al., 2011).

c. Jessica Reif, a media analyst at Merrill Lynch & Co., said, "If they can get up and running with exclusive programming within six months, it doesn't set the venture back that far."[9]

6. Run the examples above through the online version of a named entity recognition tagger, such as the Allen NLP system here: http://demo.allennlp.org/named-entity-recognition. Do the predicted tags match your annotations?

7. Build a whitespace tokenizer for English:

a) Using the NLTK library, download the complete text to the novel *Alice in Wonderland* (Carroll, 1865). Hold out the final 1000 words as a test set.

b) Label each alphanumeric character as a segmentation point, $y_m = 1$ if m is the final character of a token. Label every other character as $y_m = 0$. Then concatenate all the tokens in the training and test sets.Make sure that the number of labels $\{y_m\}_{m=1}^{M}$ is identical to the number of characters $\{c_m\}_{m=1}^{M}$ in your concatenated datasets.

c) Train a logistic regression classifier to predict y_m, using the surrounding characters $\boldsymbol{c}_{m-5:m+5}$ as features. After training the classifier, run it on the test set, using the predicted segmentation points to re-tokenize the text.

[9] From the Message Understanding Conference (MUC-7) dataset (Chinchor and Robinson, 1997).

d) Compute the per-character segmentation accuracy on the test set. You should be able to get at least 88% accuracy.

e) Print out a sample of segmented text from the test set, e.g.

```
Thereareno mice in the air , I ' m afraid , but y oumight cat
   chabat , and that ' svery like a mouse , youknow . But
   docatseat bats , I wonder ?'
```

8. Perform the following extensions to your tokenizer in the previous problem.

a) Train a conditional random field sequence labeler, by incorporating the tag bigrams (y_{m-1}, y_m) as additional features. You may use a structured prediction library such as CRFSuite, or you may want to implement Viterbi yourself. Compare the accuracy with your classification-based approach.

b) Compute the token-level performance: treating the original tokenization as ground truth, compute the number of true positives (tokens that are in both the ground truth and predicted tokenization), false positives (tokens that are in the predicted tokenization but not the ground truth), and false negatives (tokens that are in the ground truth but not the predicted tokenization). Compute the F-measure.

Hint: to match predicted and ground truth tokens, add "anchors" for the start character of each token. The number of true positives is then the size of the intersection of the sets of predicted and ground truth tokens.

c) Apply the same methodology in a more practical setting: tokenization of Chinese, which is written without whitespace. You can find annotated datasets at `http://alias-i.com/lingpipe/demos/tutorial/chineseTokens/read-me.html`.