CHAPTER

# 8

# Sequence Labeling for Parts of Speech and Named Entities

To each word a warbling note
*A Midsummer Night's Dream*, V.I

Dionysius Thrax of Alexandria (*c.* 100 B.C.), or perhaps someone else (it was a long time ago), wrote a grammatical sketch of Greek (a "*technē*") that summarized the linguistic knowledge of his day. This work is the source of an astonishing proportion of modern linguistic vocabulary, including the words *syntax*, *diphthong*, *clitic*, and *analogy*. Also included are a description of eight **parts of speech**: noun, verb, pronoun, preposition, adverb, conjunction, participle, and article. Although earlier scholars (including Aristotle as well as the Stoics) had their own lists of parts of speech, it was Thrax's set of eight that became the basis for descriptions of European languages for the next 2000 years. (All the way to the *Schoolhouse Rock* educational television shows of our childhood, which had songs about 8 parts of speech, like the late great Bob Dorough's *Conjunction Junction.*) The durability of parts of speech through two millennia speaks to their centrality in models of human language.

**parts of speech**

Proper names are another important and anciently studied linguistic category. While parts of speech are generally assigned to individual words or morphemes, a proper name is often an entire multiword phrase, like the name "Marie Curie", the location "New York City", or the organization "Stanford University". We'll use the term **named entity** for, roughly speaking, anything that can be referred to with a proper name: a person, a location, an organization, although as we'll see the term is commonly extended to include things that aren't entities per se.

**named entity**

**POS**

Parts of speech (also known as **POS**) and named entities are useful clues to sentence structure and meaning. Knowing whether a word is a noun or a verb tells us about likely neighboring words (nouns in English are preceded by determiners and adjectives, verbs by nouns) and syntactic structure (verbs have dependency links to nouns), making part-of-speech tagging a key aspect of parsing. Knowing if a named entity like *Washington* is a name of a person, a place, or a university is important to many natural language processing tasks like question answering, stance detection, or information extraction.

In this chapter we'll introduce the task of **part-of-speech tagging**, taking a sequence of words and assigning each word a part of speech like NOUN or VERB, and the task of **named entity recognition** (**NER**), assigning words or phrases tags like PERSON, LOCATION, or ORGANIZATION.

Such tasks in which we assign, to each word $x_i$ in an input word sequence, a label $y_i$, so that the output sequence $Y$ has the same length as the input sequence $X$ are called **sequence labeling** tasks. We'll introduce classic sequence labeling algorithms, one generative— the Hidden Markov Model (HMM)—and one discriminative—the Conditional Random Field (CRF). In following chapters we'll introduce modern sequence labelers based on RNNs and Transformers.

**sequence labeling**

## 8.1 (Mostly) English Word Classes

Until now we have been using part-of-speech terms like **noun** and **verb** rather freely. In this section we give more complete definitions. While word classes do have semantic tendencies—adjectives, for example, often describe *properties* and nouns *people*— parts of speech are defined instead based on their grammatical relationship with neighboring words or the morphological properties about their affixes.

| | Tag | Description | Example |
|---|---|---|---|
| Open Class | **ADJ** | Adjective: noun modifiers describing properties | *red*, *young*, *awesome* |
| | **ADV** | Adverb: verb modifiers of time, place, manner | *very*, *slowly*, *home*, *yesterday* |
| | **NOUN** | words for persons, places, things, etc. | *algorithm*, *cat*, *mango*, *beauty* |
| | **VERB** | words for actions and processes | *draw*, *provide*, *go* |
| | **PROPN** | Proper noun: name of a person, organization, place, etc.. | *Regina*, *IBM*, *Colorado* |
| | **INTJ** | Interjection: exclamation, greeting, yes/no response, etc. | *oh*, *um*, *yes*, *hello* |
| Closed Class Words | **ADP** | Adposition (Preposition/Postposition): marks a noun's spacial, temporal, or other relation | *in, on, by, under* |
| | **AUX** | Auxiliary: helping verb marking tense, aspect, mood, etc., | *can, may, should, are* |
| | **CCONJ** | Coordinating Conjunction: joins two phrases/clauses | *and*, *or*, *but* |
| | **DET** | Determiner: marks noun phrase properties | *a, an, the, this* |
| | **NUM** | Numeral | *one, two, first, second* |
| | **PART** | Particle: a preposition-like form used together with a verb | *up, down, on, off, in, out, at, by* |
| | **PRON** | Pronoun: a shorthand for referring to an entity or event | *she, who, I, others* |
| | **SCONJ** | Subordinating Conjunction: joins a main clause with a subordinate clause such as a sentential complement | *that*, *which* |
| Other | **PUNCT** | Punctuation | ; , () |
| | **SYM** | Symbols like $ or emoji | $, % |
| | **X** | Other | asdf, qwfg |

**Figure 8.1** The 17 parts of speech in the Universal Dependencies tagset (Nivre et al., 2016a). Features can be added to make finer-grained distinctions (with properties like number, case, definiteness, and so on).

**closed class**
**open class**
**function word**

Parts of speech fall into two broad categories: **closed class** and **open class**. Closed classes are those with relatively fixed membership, such as prepositions—new prepositions are rarely coined. By contrast, nouns and verbs are open classes—new nouns and verbs like *iPhone* or *to fax* are continually being created or borrowed. Closed class words are generally **function words** like *of*, *it*, *and*, or *you*, which tend to be very short, occur frequently, and often have structuring uses in grammar.

Four major open classes occur in the languages of the world: **nouns** (including proper nouns), **verbs**, **adjectives**, and **adverbs**, as well as the smaller open class of **interjections**. English has all five, although not every language does.

**noun**
**common noun**
**count noun**
**mass noun**
**proper noun**

**Nouns** are words for people, places, or things, but include others as well. **Common nouns** include concrete terms like *cat* and *mango*, abstractions like *algorithm* and *beauty*, and verb-like terms like *pacing* as in *His pacing to and fro became quite annoying*. Nouns in English can occur with determiners (*a goat, this bandwidth*) take possessives (*IBM's annual revenue*), and may occur in the plural (*goats, abaci*). Many languages, including English, divide common nouns into **count nouns** and **mass nouns**. Count nouns can occur in the singular and plural (*goat/goats, relationship/relationships*) and can be counted (*one goat, two goats*). Mass nouns are used when something is conceptualized as a homogeneous group. So *snow, salt*, and *communism* are not counted (i.e., **two snows* or **two communisms*). **Proper nouns**, like *Regina*, *Colorado*, and *IBM*, are names of specific persons or entities.

**verb**

**Verbs** refer to actions and processes, including main verbs like *draw*, *provide*, and *go*. English verbs have inflections (non-third-person-singular (*eat*), third-person-singular (*eats*), progressive (*eating*), past participle (*eaten*)). While many scholars believe that all human languages have the categories of noun and verb, others have argued that some languages, such as Riau Indonesian and Tongan, don't even make this distinction (Broschart 1997; Evans 2000; Gil 2000) .

**adjective**

**Adjectives** often describe properties or qualities of nouns, like color (*white*, *black*), age (*old*, *young*), and value (*good*, *bad*), but there are languages without adjectives. In Korean, for example, the words corresponding to English adjectives act as a subclass of verbs, so what is in English an adjective "beautiful" acts in Korean like a verb meaning "to be beautiful".

**adverb**

**Adverbs** are a hodge-podge. All the italicized words in this example are adverbs:

> *Actually*, I ran *home extremely quickly yesterday*

**locative**
**degree**
**manner**
**temporal**

Adverbs generally modify something (often verbs, hence the name "adverb", but also other adverbs and entire verb phrases). **Directional adverbs** or **locative adverbs** (*home*, *here*, *downhill*) specify the direction or location of some action; **degree adverbs** (*extremely*, *very*, *somewhat*) specify the extent of some action, process, or property; **manner adverbs** (*slowly*, *slinkily*, *delicately*) describe the manner of some action or process; and **temporal adverbs** describe the time that some action or event took place (*yesterday*, *Monday*).

**interjection**

**Interjections** (*oh, hey, alas, uh, um*) are a smaller open class that also includes greetings (*hello, goodbye*) and question responses (*yes, no, uh-huh*).

**preposition**

English adpositions occur before nouns, hence are called **prepositions**. They can indicate spatial or temporal relations, whether literal (*on it*, *before then*, *by the house*) or metaphorical (*on time*, *with gusto*, *beside herself*), and relations like marking the agent in *Hamlet was written by Shakespeare*.

**particle**
**phrasal verb**

A **particle** resembles a preposition or an adverb and is used in combination with a verb. Particles often have extended meanings that aren't quite the same as the prepositions they resemble, as in the particle *over* in *she turned the paper over*. A verb and a particle acting as a single unit is called a **phrasal verb**. The meaning of phrasal verbs is often **non-compositional**—not predictable from the individual meanings of the verb and the particle. Thus, *turn down* means 'reject', *rule out* 'eliminate', and *go on* 'continue'.

**determiner**
**article**

**Determiners** like *this* and *that* (*this chapter*, *that page*) can mark the start of an English noun phrase. **Articles** like *a*, *an*, and *the*, are a type of determiner that mark discourse properties of the noun and are quite frequent; *the* is the most common word in written English, with *a* and *an* right behind.

**conjunction**
**complementizer**

**Conjunctions** join two phrases, clauses, or sentences. Coordinating conjunctions like *and*, *or*, and *but* join two elements of equal status. Subordinating conjunctions are used when one of the elements has some embedded status. For example, the subordinating conjunction *that* in *"I thought that you might like some milk"* links the main clause *I thought* with the subordinate clause *you might like some milk*. This clause is called subordinate because this entire clause is the "content" of the main verb *thought*. Subordinating conjunctions like *that* which link a verb to its argument in this way are also called **complementizers**.

**pronoun**
**wh**

**Pronouns** act as a shorthand for referring to an entity or event. Personal pronouns refer to persons or entities (*you*, *she*, *I*, *it*, *me*, etc.). Possessive pronouns are forms of personal pronouns that indicate either actual possession or more often just an abstract relation between the person and some object (*my, your, his, her, its, one's, our, their*). **Wh-pronouns** (*what, who, whom, whoever*) are used in certain question forms, or act as complementizers (*Frida, who married Diego...*).

auxiliary

**Auxiliary** verbs mark semantic features of a main verb such as its tense, whether it is completed (aspect), whether it is negated (polarity), and whether an action is necessary, possible, suggested, or desired (mood). English auxiliaries include the **copula** verb *be*, the two verbs *do* and *have*, forms, as well as **modal verbs** used to mark the mood associated with the event depicted by the main verb: *can* indicates ability or possibility, *may* permission or possibility, *must* necessity.

copula

modal

An English-specific tagset, the 45-tag Penn Treebank tagset (Marcus et al., 1993), shown in Fig. 8.2, has been used to label many syntactically annotated corpora like the Penn Treebank corpora, so is worth knowing about.

| Tag | Description | Example | Tag | Description | Example | Tag | Description | Example |
|---|---|---|---|---|---|---|---|---|
| CC | coord. conj. | *and, but, or* | NNP | proper noun, sing. | *IBM* | TO | "to" | *to* |
| CD | cardinal number | *one, two* | NNPS | proper noun, plu. | *Carolinas* | UH | interjection | *ah, oops* |
| DT | determiner | *a, the* | NNS | noun, plural | *llamas* | VB | verb base | *eat* |
| EX | existential 'there' | *there* | PDT | predeterminer | *all, both* | VBD | verb past tense | *ate* |
| FW | foreign word | *mea culpa* | POS | possessive ending | *'s* | VBG | verb gerund | *eating* |
| IN | preposition/ subordin-conj | *of, in, by* | PRP | personal pronoun | *I, you, he* | VBN | verb past participle | *eaten* |
| JJ | adjective | *yellow* | PRP$ | possess. pronoun | *your, one's* | VBP | verb non-3sg-pr | *eat* |
| JJR | comparative adj | *bigger* | RB | adverb | *quickly* | VBZ | verb 3sg pres | *eats* |
| JJS | superlative adj | *wildest* | RBR | comparative adv | *faster* | WDT | wh-determ. | *which, that* |
| LS | list item marker | *1, 2, One* | RBS | superlatv. adv | *fastest* | WP | wh-pronoun | *what, who* |
| MD | modal | *can, should* | RP | particle | *up, off* | WP$ | wh-possess. | *whose* |
| NN | sing or mass noun | *llama* | SYM | symbol | *+,%, &* | WRB | wh-adverb | *how, where* |

**Figure 8.2** Penn Treebank part-of-speech tags.

Below we show some examples with each word tagged according to both the UD and Penn tagsets. Notice that the Penn tagset distinguishes tense and participles on verbs, and has a special tag for the existential *there* construction in English. Note that since *New England Journal of Medicine* is a proper noun, both tagsets mark its component nouns as NNP, including *journal* and *medicine*, which might otherwise be labeled as common nouns (NOUN/NN).

(8.1) There/PRO/EX are/VERB/VBP 70/NUM/CD children/NOUN/NNS there/ADV/RB ./PUNC/.

(8.2) Preliminary/ADJ/JJ findings/NOUN/NNS were/AUX/VBD reported/VERB/VBN in/ADP/IN today/NOUN/NN 's/PART/POS New/PROPN/NNP England/PROPN/NNP Journal/PROPN/NNP of/ADP/IN Medicine/PROPN/NNP

## 8.2 Part-of-Speech Tagging

part-of-speech tagging

**Part-of-speech tagging** is the process of assigning a part-of-speech to each word in a text. The input is a sequence $x_1, x_2, ..., x_n$ of (tokenized) words and a tagset, and the output is a sequence $y_1, y_2, ..., y_n$ of tags, each output $y_i$ corresponding exactly to one input $x_i$, as shown in the intuition in Fig. 8.3.

ambiguous

Tagging is a **disambiguation** task; words are **ambiguous** —have more than one possible part-of-speech—and the goal is to find the correct tag for the situation. For example, *book* can be a verb (*book that flight*) or a noun (*hand me that book*). *That* can be a determiner (*Does that flight serve dinner*) or a complementizer (*I*

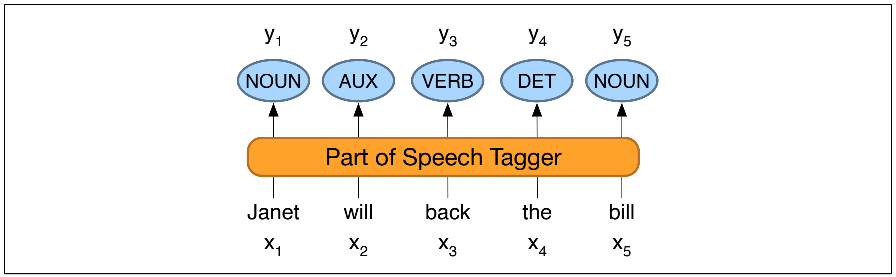

**Figure 8.3** The task of part-of-speech tagging: mapping from input words $x_1, x_2, \ldots, x_n$ to output POS tags $y_1, y_2, \ldots, y_n$ .

**ambiguity resolution**

*thought that your flight was earlier*). The goal of POS-tagging is to **resolve** these ambiguities, choosing the proper tag for the context.

**accuracy**

The **accuracy** of part-of-speech tagging algorithms (the percentage of test set tags that match human gold labels) is extremely high. One study found accuracies over 97% across 15 languages from the Universal Dependency (UD) treebank (Wu and Dredze, 2019). Accuracies on various English treebanks are also 97% (no matter the algorithm; HMMs, CRFs, BERT perform similarly). This 97% number is also about the human performance on this task, at least for English (Manning, 2011).

| **Types:** | | **WSJ** | | **Brown** | |
|---|---|---|---|---|---|
| **Unambiguous** | (1 tag) | 44,432 | (**86%**) | 45,799 | (**85%**) |
| **Ambiguous** | (2+ tags) | 7,025 | (**14%**) | 8,050 | (**15%**) |
| **Tokens**: | | | | | |
| **Unambiguous** | (1 tag) | 577,421 | (**45%**) | 384,349 | (**33%**) |
| **Ambiguous** | (2+ tags) | 711,780 | (**55%**) | 786,646 | (**67%**) |

**Figure 8.4** Tag ambiguity in the Brown and WSJ corpora (Treebank-3 45-tag tagset).

We'll introduce algorithms for the task in the next few sections, but first let's explore the task. Exactly how hard is it? Fig. 8.4 shows that most word types (85-86%) are unambiguous (*Janet* is always NNP, *hesitantly* is always RB). But the ambiguous words, though accounting for only 14-15% of the vocabulary, are very common, and 55-67% of word tokens in running text are ambiguous. Particularly ambiguous common words include *that*, *back*, *down*, *put* and *set*; here are some examples of the 6 different parts of speech for the word *back*:

earnings growth took a **back/JJ** seat
a small building in the **back/NN**
a clear majority of senators **back/VBP** the bill
Dave began to **back/VB** toward the door
enable the country to buy **back/RP** debt
I was twenty-one **back/RB** then

Nonetheless, many words are easy to disambiguate, because their different tags aren't equally likely. For example, *a* can be a determiner or the letter *a*, but the determiner sense is much more likely.

This idea suggests a useful **baseline**: given an ambiguous word, choose the tag which is **most frequent** in the training corpus. This is a key concept:

**Most Frequent Class Baseline:** Always compare a classifier against a baseline at least as good as the most frequent class baseline (assigning each token to the class it occurred in most often in the training set).

The most-frequent-tag baseline has an accuracy of about 92%[1]. The baseline thus differs from the state-of-the-art and human ceiling (97%) by only 5%.

## 8.3 Named Entities and Named Entity Tagging

Part of speech tagging can tell us that words like *Janet*, *Stanford University*, and *Colorado* are all proper nouns; being a proper noun is a grammatical property of these words. But viewed from a semantic perspective, these proper nouns refer to different kinds of entities: Janet is a person, Stanford University is an organization, and Colorado is a location.

**named entity**

**named entity recognition**

**NER**

A **named entity** is, roughly speaking, anything that can be referred to with a proper name: a person, a location, an organization. The task of **named entity recognition** (**NER**) is to find spans of text that constitute proper names and tag the type of the entity. Four entity tags are most common: **PER** (person), **LOC** (location), **ORG** (organization), or **GPE** (geo-political entity). However, the term **named entity** is commonly extended to include things that aren't entities per se, including dates, times, and other kinds of temporal expressions, and even numerical expressions like prices. Here's an example of the output of an NER tagger:

> Citing high fuel prices, [$_{\text{ORG}}$ **United Airlines**] said [$_{\text{TIME}}$ **Friday**] it has increased fares by [$_{\text{MONEY}}$ **$6**] per round trip on flights to some cities also served by lower-cost carriers. [$_{\text{ORG}}$ **American Airlines**], a unit of [$_{\text{ORG}}$ **AMR Corp.**], immediately matched the move, spokesman [$_{\text{PER}}$ **Tim Wagner**] said. [$_{\text{ORG}}$ **United**], a unit of [$_{\text{ORG}}$ **UAL Corp.**], said the increase took effect [$_{\text{TIME}}$ **Thursday**] and applies to most routes where it competes against discount carriers, such as [$_{\text{LOC}}$ **Chicago**] to [$_{\text{LOC}}$ **Dallas**] and [$_{\text{LOC}}$ **Denver**] to [$_{\text{LOC}}$ **San Francisco**].

The text contains 13 mentions of named entities including 5 organizations, 4 locations, 2 times, 1 person, and 1 mention of money. Figure 8.5 shows typical generic named entity types. Many applications will also need to use specific entity types like proteins, genes, commercial products, or works of art.

| Type | Tag | Sample Categories | Example sentences |
|---|---|---|---|
| People | PER | people, characters | **Turing** is a giant of computer science. |
| Organization | ORG | companies, sports teams | The **IPCC** warned about the cyclone. |
| Location | LOC | regions, mountains, seas | **Mt. Sanitas** is in **Sunshine Canyon**. |
| Geo-Political Entity | GPE | countries, states | **Palo Alto** is raising the fees for parking. |

**Figure 8.5** A list of generic named entity types with the kinds of entities they refer to.

Named entity tagging is a useful first step in lots of natural language processing tasks. In sentiment analysis we might want to know a consumer's sentiment toward a particular entity. Entities are a useful first stage in question answering, or for linking text to information in structured knowledge sources like Wikipedia. And named entity tagging is also central to tasks involving building semantic representations, like extracting events and the relationship between participants.

Unlike part-of-speech tagging, where there is no segmentation problem since each word gets one tag, the task of named entity recognition is to find and label *spans* of text, and is difficult partly because of the ambiguity of segmentation; we

---

[1] In English, on the WSJ corpus, tested on sections 22-24.

need to decide what's an entity and what isn't, and where the boundaries are. Indeed, most words in a text will not be named entities. Another difficulty is caused by type ambiguity. The mention `JFK` can refer to a person, the airport in New York, or any number of schools, bridges, and streets around the United States. Some examples of this kind of cross-type confusion are given in Figure 8.6.

$[_{\text{PER}}$ Washington] was born into slavery on the farm of James Burroughs.
$[_{\text{ORG}}$ Washington] went up 2 games to 1 in the four-game series.
Blair arrived in $[_{\text{LOC}}$ Washington] for what may well be his last state visit.
In June, $[_{\text{GPE}}$ Washington] passed a primary seatbelt law.

**Figure 8.6** Examples of type ambiguities in the use of the name *Washington*.

The standard approach to sequence labeling for a span-recognition problem like NER is **BIO** tagging (Ramshaw and Marcus, 1995). This is a method that allows us to treat NER like a word-by-word sequence labeling task, via tags that capture both the boundary and the named entity type. Consider the following sentence:

> $[_{\text{PER}}$ **Jane Villanueva** ] of $[_{\text{ORG}}$ **United**] , a unit of $[_{\text{ORG}}$ **United Airlines Holding**] , said the fare applies to the $[_{\text{LOC}}$ **Chicago** ] route.

**BIO**

Figure 8.7 shows the same excerpt represented with **BIO** tagging, as well as variants called **IO** tagging and **BIOES** tagging. In BIO tagging we label any token that *begins* a span of interest with the label B, tokens that occur *inside* a span are tagged with an I, and any tokens outside of any span of interest are labeled O. While there is only one O tag, we'll have distinct B and I tags for each named entity class. The number of tags is thus $2n+1$ tags, where $n$ is the number of entity types. BIO tagging can represent exactly the same information as the bracketed notation, but has the advantage that we can represent the task in the same simple sequence modeling way as part-of-speech tagging: assigning a single label $y_i$ to each input word $x_i$:

| Words | IO Label | BIO Label | BIOES Label |
|---|---|---|---|
| Jane | I-PER | B-PER | B-PER |
| Villanueva | I-PER | I-PER | E-PER |
| of | O | O | O |
| United | I-ORG | B-ORG | B-ORG |
| Airlines | I-ORG | I-ORG | I-ORG |
| Holding | I-ORG | I-ORG | E-ORG |
| discussed | O | O | O |
| the | O | O | O |
| Chicago | I-LOC | B-LOC | S-LOC |
| route | O | O | O |
| . | O | O | O |

**Figure 8.7** NER as a sequence model, showing IO, BIO, and BIOES taggings.

We've also shown two variant tagging schemes: IO tagging, which loses some information by eliminating the B tag, and BIOES tagging, which adds an end tag E for the end of a span, and a span tag S for a span consisting of only one word. A sequence labeler (HMM, CRF, RNN, Transformer, etc.) is trained to label each token in a text with tags that indicate the presence (or absence) of particular kinds of named entities.

## 8.4 HMM Part-of-Speech Tagging

## 8.5 Conditional Random Fields (CRFs)

## 8.6 Evaluation of Named Entity Recognition

Part-of-speech taggers are evaluated by the standard metric of **accuracy**. Named entity recognizers are evaluated by **recall**, **precision**, and **F$_1$ measure**. Recall that recall is the ratio of the number of correctly labeled responses to the total that should have been labeled; precision is the ratio of the number of correctly labeled responses to the total labeled; and $F$-measure is the harmonic mean of the two.

To know if the difference between the F$_1$ scores of two NER systems is a significant difference, we use the paired bootstrap test, or the similar randomization test (Section **??**).

For named entity tagging, the *entity* rather than the word is the unit of response. Thus in the example in Fig. 8.16, the two entities *Jane Villanueva* and *United Airlines Holding* and the non-entity *discussed* would each count as a single response.

The fact that named entity tagging has a segmentation component which is not present in tasks like text categorization or part-of-speech tagging causes some problems with evaluation. For example, a system that labeled *Jane* but not *Jane Villanueva* as a person would cause two errors, a false positive for O and a false negative for I-PER. In addition, using entities as the unit of response but words as the unit of training means that there is a mismatch between the training and test conditions.

## 8.7 Further Details

## 8.8 Summary

This chapter introduced **parts of speech** and **named entities**, and the tasks of **part-of-speech tagging** and **named entity recognition**:

- Languages generally have a small set of **closed class** words that are highly frequent, ambiguous, and act as **function words**, and **open-class** words like **nouns**, **verbs**, **adjectives**. Various part-of-speech **tagsets** exist, of between 40 and 200 tags.
- **Part-of-speech tagging** is the process of assigning a part-of-speech label to each of a sequence of words.
- **Named entities** are words for proper nouns referring mainly to people, places, and organizations, but extended to many other types that aren't strictly entities or even proper nouns.
- Two common approaches to **sequence modeling** are a **generative** approach, **HMM** tagging, and a **discriminative** approach, **CRF** tagging. We will see a neural approach in following chapters.
- The probabilities in HMM taggers are estimated by maximum likelihood estimation on tag-labeled training corpora. The Viterbi algorithm is used for **decoding**, finding the most likely tag sequence
- **Conditional Random Fields** or **CRF taggers** train a log-linear model that can choose the best tag sequence given an observation sequence, based on features that condition on the output tag, the prior output tag, the entire input sequence, and the current timestep. They use the Viterbi algorithm for inference, to choose the best sequence of tags, and a version of the Forward-Backward algorithm (see Appendix A) for training,

# Bibliographical and Historical Notes

What is probably the earliest part-of-speech tagger was part of the parser in Zellig Harris's Transformations and Discourse Analysis Project (TDAP), implemented between June 1958 and July 1959 at the University of Pennsylvania (Harris, 1962), although earlier systems had used part-of-speech dictionaries. TDAP used 14 handwritten rules for part-of-speech disambiguation; the use of part-of-speech tag sequences and the relative frequency of tags for a word prefigures modern algorithms. The parser was implemented essentially as a cascade of finite-state transducers; see Joshi and Hopely (1999) and Karttunen (1999) for a reimplementation.

The Computational Grammar Coder (CGC) of Klein and Simmons (1963) had three components: a lexicon, a morphological analyzer, and a context disambiguator. The small 1500-word lexicon listed only function words and other irregular words. The morphological analyzer used inflectional and derivational suffixes to assign part-of-speech classes. These were run over words to produce candidate parts of speech which were then disambiguated by a set of 500 context rules by relying on surrounding islands of unambiguous words. For example, one rule said that between an ARTICLE and a VERB, the only allowable sequences were ADJ-NOUN, NOUN-ADVERB, or NOUN-NOUN. The TAGGIT tagger (Greene and Rubin, 1971) used the same architecture as Klein and Simmons (1963), with a bigger dictionary and more tags (87). TAGGIT was applied to the Brown corpus and, according to Francis and Kučera (1982, p. 9), accurately tagged 77% of the corpus; the remainder of the Brown corpus was then tagged by hand. All these early algorithms were based on a two-stage architecture in which a dictionary was first used to assign each word a set of potential parts of speech, and then lists of handwritten disambiguation rules winnowed the set down to a single part of speech per word.

Probabilities were used in tagging by Stolz et al. (1965) and a complete probabilistic tagger with Viterbi decoding was sketched by Bahl and Mercer (1976). The Lancaster-Oslo/Bergen (LOB) corpus, a British English equivalent of the Brown corpus, was tagged in the early 1980's with the CLAWS tagger (Marshall 1983; Marshall 1987; Garside 1987), a probabilistic algorithm that approximated a simplified HMM tagger. The algorithm used tag bigram probabilities, but instead of storing the word likelihood of each tag, the algorithm marked tags either as *rare* ($P(\text{tag}|\text{word}) < .01$) *infrequent* ($P(\text{tag}|\text{word}) < .10$) or *normally frequent* ($P(\text{tag}|\text{word}) > .10$).

DeRose (1988) developed a quasi-HMM algorithm, including the use of dynamic programming, although computing $P(t|w)P(w)$ instead of $P(w|t)P(w)$. The same year, the probabilistic PARTS tagger of Church 1988, 1989 was probably the first implemented HMM tagger, described correctly in Church (1989), although Church (1988) also described the computation incorrectly as $P(t|w)P(w)$ instead of $P(w|t)P(w)$. Church (p.c.) explained that he had simplified for pedagogical purposes because using the probability $P(t|w)$ made the idea seem more understandable as "storing a lexicon in an almost standard form".

Later taggers explicitly introduced the use of the hidden Markov model (Kupiec 1992; Weischedel et al. 1993; Schütze and Singer 1994). Merialdo (1994) showed that fully unsupervised EM didn't work well for the tagging task and that reliance on hand-labeled data was important. Charniak et al. (1993) showed the importance of the most frequent tag baseline; the 92.3% number we give above was from Abney et al. (1999). See Brants (2000) for HMM tagger implementation details, including the extension to trigram contexts, and the use of sophisticated unknown word features; its performance is still close to state of the art taggers.

Log-linear models for POS tagging were introduced by Ratnaparkhi (1996), who introduced a system called MXPOST which implemented a maximum entropy Markov model (MEMM), a slightly simpler version of a CRF. Around the same time, sequence labelers were applied to the task of named entity tagging, first with HMMs (Bikel et al., 1997) and MEMMs (McCallum et al., 2000), and then once CRFs were developed (Lafferty et al. 2001), they were also applied to NER (McCallum and Li, 2003). A wide exploration of features followed (Zhou et al., 2005). Neural approaches to NER mainly follow from the pioneering results of Collobert et al. (2011), who applied a CRF on top of a convolutional net. BiLSTMs with word and character-based embeddings as input followed shortly and became a standard neural algorithm for NER (Huang et al. 2015, Ma and Hovy 2016, Lample et al. 2016) followed by the more recent use of Transformers and BERT.

The idea of using letter suffixes for unknown words is quite old; the early Klein and Simmons (1963) system checked all final letter suffixes of lengths 1-5. The unknown word features described on page 17 come mainly from Ratnaparkhi (1996), with augmentations from Toutanova et al. (2003) and Manning (2011).

State of the art POS taggers use neural algorithms, either bidirectional RNNs or Transformers like BERT; see Chapter 9 and Chapter 11. HMM (Brants 2000; Thede and Harper 1999) and CRF tagger accuracies are likely just a tad lower.

Manning (2011) investigates the remaining 2.7% of errors in a high-performing tagger (Toutanova et al., 2003). He suggests that a third or half of these remaining errors are due to errors or inconsistencies in the training data, a third might be solvable with richer linguistic models, and for the remainder the task is underspecified or unclear.

Supervised tagging relies heavily on in-domain training data hand-labeled by experts. Ways to relax this assumption include unsupervised algorithms for clustering words into part-of-speech-like classes, summarized in Christodoulopoulos et al. (2010), and ways to combine labeled and unlabeled data, for example by co-training (Clark et al. 2003; Søgaard 2010).

See Householder (1995) for historical notes on parts of speech, and Sampson (1987) and Garside et al. (1997) on the provenance of the Brown and other tagsets.

# Exercises

**8.1** Find one tagging error in each of the following sentences that are tagged with the Penn Treebank tagset:

1. I/PRP need/VBP a/DT flight/NN from/IN Atlanta/NN
2. Does/VBZ this/DT flight/NN serve/VB dinner/NNS
3. I/PRP have/VB a/DT friend/NN living/VBG in/IN Denver/NNP
4. Can/VBP you/PRP list/VB the/DT nonstop/JJ afternoon/NN flights/NNS

**8.2** Use the Penn Treebank tagset to tag each word in the following sentences from Damon Runyon's short stories. You may ignore punctuation. Some of these are quite difficult; do your best.

1. It is a nice night.
2. This crap game is over a garage in Fifty-second Street. . .
3. . . . Nobody ever takes the newspapers she sells . . .
4. He is a tall, skinny guy with a long, sad, mean-looking kisser, and a mournful voice.

5. ...I am sitting in Mindy's restaurant putting on the gefillte fish, which is a dish I am very fond of, ...
6. When a guy and a doll get to taking peeks back and forth at each other, why there you are indeed.

**8.3** Now compare your tags from the previous exercise with one or two friend's answers. On which words did you disagree the most? Why?

**8.4** Implement the "most likely tag" baseline. Find a POS-tagged training set, and use it to compute for each word the tag that maximizes $p(t|w)$. You will need to implement a simple tokenizer to deal with sentence boundaries. Start by assuming that all unknown words are NN and compute your error rate on known and unknown words. Now write at least five rules to do a better job of tagging unknown words, and show the difference in error rates.

**8.5** Build a bigram HMM tagger. You will need a part-of-speech-tagged corpus. First split the corpus into a training set and test set. From the labeled training set, train the transition and observation probabilities of the HMM tagger directly on the hand-tagged data. Then implement the Viterbi algorithm so you can decode a test sentence. Now run your algorithm on the test set. Report its error rate and compare its performance to the most frequent tag baseline.

**8.6** Do an error analysis of your tagger. Build a confusion matrix and investigate the most frequent errors. Propose some features for improving the performance of your tagger on these errors.

**8.7** Develop a set of regular expressions to recognize the character shape features described on page 17.

**8.8** The BIO and other labeling schemes given in this chapter aren't the only possible one. For example, the B tag can be reserved only for those situations where an ambiguity exists between adjacent entities. Propose a new set of BIO tags for use with your NER system. Experiment with it and compare its performance with the schemes presented in this chapter.

**8.9** Names of works of art (books, movies, video games, etc.) are quite different from the kinds of named entities we've discussed in this chapter. Collect a list of names of works of art from a particular category from a Web-based source (e.g., gutenberg.org, amazon.com, imdb.com, etc.). Analyze your list and give examples of ways that the names in it are likely to be problematic for the techniques described in this chapter.

**8.10** Develop an NER system specific to the category of names that you collected in the last exercise. Evaluate your system on a collection of text likely to contain instances of these named entities.

Abney, S. P., R. E. Schapire, and Y. Singer. 1999. Boosting applied to tagging and PP attachment. *EMNLP/VLC*.

Bada, M., M. Eckert, D. Evans, K. Garcia, K. Shipley, D. Sitnikov, W. A. Baumgartner, K. B. Cohen, K. Verspoor, J. A. Blake, and L. E. Hunter. 2012. Concept annotation in the craft corpus. *BMC bioinformatics*, 13(1):161.

Bahl, L. R. and R. L. Mercer. 1976. Part of speech assignment by a statistical decision algorithm. *Proceedings IEEE International Symposium on Information Theory*.

Bamman, D., S. Popat, and S. Shen. 2019. An annotated dataset of literary entities. *NAACL HLT*.

Bikel, D. M., S. Miller, R. Schwartz, and R. Weischedel. 1997. Nymble: A high-performance learning name-finder. *ANLP*.

Brants, T. 2000. TnT: A statistical part-of-speech tagger. *ANLP*.

Broschart, J. 1997. Why Tongan does it differently. *Linguistic Typology*, 1:123–165.

Charniak, E., C. Hendrickson, N. Jacobson, and M. Perkowitz. 1993. Equations for part-of-speech tagging. *AAAI*.

Chiticariu, L., M. Danilevsky, Y. Li, F. Reiss, and H. Zhu. 2018. SystemT: Declarative text understanding for enterprise. *NAACL HLT*, volume 3.

Chiticariu, L., Y. Li, and F. R. Reiss. 2013. Rule-Based Information Extraction is Dead! Long Live Rule-Based Information Extraction Systems! *EMNLP*.

Christodoulopoulos, C., S. Goldwater, and M. Steedman. 2010. Two decades of unsupervised POS induction: How far have we come? *EMNLP*.

Church, K. W. 1988. A stochastic parts program and noun phrase parser for unrestricted text. *ANLP*.

Church, K. W. 1989. A stochastic parts program and noun phrase parser for unrestricted text. *ICASSP*.

Clark, S., J. R. Curran, and M. Osborne. 2003. Bootstrapping POS-taggers using unlabelled data. *CoNLL*.

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537.

DeRose, S. J. 1988. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14:31–39.

Evans, N. 2000. Word classes in the world's languages. In G. Booij, C. Lehmann, and J. Mugdan, editors, *Morphology: A Handbook on Inflection and Word Formation*, pages 708–732. Mouton.

Francis, W. N. and H. Kučera. 1982. *Frequency Analysis of English Usage*. Houghton Mifflin, Boston.

Garside, R. 1987. The CLAWS word-tagging system. In R. Garside, G. Leech, and G. Sampson, editors, *The Computational Analysis of English*, pages 30–41. Longman.

Garside, R., G. Leech, and A. McEnery. 1997. *Corpus Annotation*. Longman.

Gil, D. 2000. Syntactic categories, cross-linguistic variation and universal grammar. In P. M. Vogel and B. Comrie, editors, *Approaches to the Typology of Word Classes*, pages 173–216. Mouton.

Greene, B. B. and G. M. Rubin. 1971. Automatic grammatical tagging of English. Department of Linguistics, Brown University, Providence, Rhode Island.

Hajič, J. 2000. Morphological tagging: Data vs. dictionaries. In *NAACL*.

Hakkani-Tür, D., K. Oflazer, and G. Tür. 2002. Statistical morphological disambiguation for agglutinative languages. *Journal of Computers and Humanities*, 36(4):381–410.

Harris, Z. S. 1962. *String Analysis of Sentence Structure*. Mouton, The Hague.

Householder, F. W. 1995. Dionysius Thrax, the *technai*, and Sextus Empiricus. In E. F. K. Koerner and R. E. Asher, editors, *Concise History of the Language Sciences*, pages 99–103. Elsevier Science.

Hovy, E. H., M. P. Marcus, M. Palmer, L. A. Ramshaw, and R. Weischedel. 2006. OntoNotes: The 90% solution. *HLT-NAACL*.

Huang, Z., W. Xu, and K. Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Joshi, A. K. and P. Hopely. 1999. A parser from antiquity. In A. Kornai, editor, *Extended Finite State Models of Language*, pages 6–15. Cambridge University Press.

Karlsson, F., A. Voutilainen, J. Heikkilä, and A. Anttila, editors. 1995. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter.

Karttunen, L. 1999. Comments on Joshi. In A. Kornai, editor, *Extended Finite State Models of Language*, pages 16–18. Cambridge University Press.

Klein, S. and R. F. Simmons. 1963. A computational approach to grammatical coding of English words. *Journal of the ACM*, 10(3):334–347.

Kupiec, J. 1992. Robust part-of-speech tagging using a hidden Markov model. *Computer Speech and Language*, 6:225–242.

Lafferty, J. D., A. McCallum, and F. C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML*.

Lample, G., M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. 2016. Neural architectures for named entity recognition. *NAACL HLT*.

Lee, H., M. Surdeanu, and D. Jurafsky. 2017. A scaffolding approach to coreference resolution integrating statistical and rule-based models. *Natural Language Engineering*, 23(5):733–762.

Ma, X. and E. H. Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. *ACL*.

Manning, C. D. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? *CICLing 2011*.

Marcus, M. P., B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330.

Marshall, I. 1983. Choice of grammatical word-class without global syntactic analysis: Tagging words in the LOB corpus. *Computers and the Humanities*, 17:139–150.

Marshall, I. 1987. Tag selection using probabilistic methods. In R. Garside, G. Leech, and G. Sampson, editors, *The Computational Analysis of English*, pages 42–56. Longman.

McCallum, A., D. Freitag, and F. C. N. Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. *ICML*.

McCallum, A. and W. Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. *CoNLL*.

Merialdo, B. 1994. Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2):155–172.

Mikheev, A., M. Moens, and C. Grover. 1999. Named entity recognition without gazetteers. *EACL*.

Nivre, J., M.-C. de Marneffe, F. Ginter, Y. Goldberg, J. Hajič, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, R. Tsarfaty, and D. Zeman. 2016a. Universal Dependencies v1: A multilingual treebank collection. *LREC*.

Nivre, J., M.-C. de Marneffe, F. Ginter, Y. Goldberg, J. Hajič, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, R. Tsarfaty, and D. Zeman. 2016b. Universal Dependencies v1: A multilingual treebank collection. *LREC*.

Oravecz, C. and P. Dienes. 2002. Efficient stochastic part-of-speech tagging for Hungarian. *LREC*.

Ramshaw, L. A. and M. P. Marcus. 1995. Text chunking using transformation-based learning. *Proceedings of the 3rd Annual Workshop on Very Large Corpora*.

Ratnaparkhi, A. 1996. A maximum entropy part-of-speech tagger. *EMNLP*.

Sampson, G. 1987. Alternative grammatical coding systems. In R. Garside, G. Leech, and G. Sampson, editors, *The Computational Analysis of English*, pages 165–183. Longman.

Schütze, H. and Y. Singer. 1994. Part-of-speech tagging using a variable memory Markov model. *ACL*.

Søgaard, A. 2010. Simple semi-supervised training of part-of-speech taggers. *ACL*.

Stolz, W. S., P. H. Tannenbaum, and F. V. Carstensen. 1965. A stochastic approach to the grammatical coding of English. *CACM*, 8(6):399–405.

Thede, S. M. and M. P. Harper. 1999. A second-order hidden Markov model for part-of-speech tagging. *ACL*.

Toutanova, K., D. Klein, C. D. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. *HLT-NAACL*.

Voutilainen, A. 1999. Handcrafted rules. In H. van Halteren, editor, *Syntactic Wordclass Tagging*, pages 217–246. Kluwer.

Weischedel, R., M. Meteer, R. Schwartz, L. A. Ramshaw, and J. Palmucci. 1993. Coping with ambiguity and unknown words through probabilistic models. *Computational Linguistics*, 19(2):359–382.

Wu, S. and M. Dredze. 2019. Beto, Bentz, Becas: The surprising cross-lingual effectiveness of BERT. *EMNLP*.

Zhou, G., J. Su, J. Zhang, and M. Zhang. 2005. Exploring various knowledge in relation extraction. *ACL*.