Contents

Chapter 1

Finding the Structure of Documents

Dilek Hakkani-Tür[1], Gokhan Tur[2], Benoit Favre[1], and Elizabeth Shriberg[2]
[1] International Computer Science Institute (ICSI), Berkeley, CA 94704
[2] SRI International, Menlo Park, CA 94025
{dilek,favre}@icsi.berkeley.edu, {gokhan,ees}@speech.sri.com

1.1 Introduction

In human language, words and sentences do not appear randomly, but usually there is a structure. For example, combinations of words form sentences - meaningful grammatical units, such as statements, requests, and commands. Likewise, in written text, sentences form paragraphs - self-contained units of discourse about a particular point or idea. Sentences may also be related to each other, by explicit discourse connectives, such as *therefore*.

Automatic extraction of structure of documents helps subsequent natural language processing (NLP) tasks; for example, parsing, machine translation and semantic role labeling use sentences as the basic processing unit [61, 57]. Sentence boundary annotation is also important for aiding human readability of the output of automatic speech recognition (ASR) systems [43]. Furthermore, chunking the input text or speech into topically coherent blocks provides better organization and indexing of the data. For example, one can wish to listen to a portion of a long speech about a specific topic. Similarly, articles belonging to the same topic may be categorized and processed further. Given the ever-growing problem of written and spoken information overload, extracting the structure of textual and audio documents is a meaningful and sometimes necessary first step in most speech and language processing applications.

Here, we discuss methods for finding the structure of documents, where only the sentence and group of sentences related to a topic are considered as the structure elements, for simplicity.

In this chapter, we call the task of deciding where sentences start and end given a sequence of characters (made of words and typographical cues) **sentence boundary detection**. Similarly, we refer to **topic segmentation** as the task of determining when a topic starts and ends in a sequence of sentences. We present statistical classification approaches that try to infer the presence of sentence and topic boundaries given human-annotated training data, for segmentation.[1] These methods base their predictions on *features* of the input: local characteristics that give evidence toward the presence or absence of a sentence or topic boundary, such as a punctuation sign, a pause in speech, and a new word in a document. Features are the core of classification approaches, and require careful design and selection in order to be successful and prevent overfitting and noise problems.

Note that while most statistical approaches described in this chapter are language independent, every language is a challenge in itself. For example, for processing of Chinese documents, one may need to first segment character sequences into words, as the words usually are not separated by a space. Similarly, for morphologically rich languages, one may need to analyze the word structure to extract additional features. Such processing is usually done in a preprocessing step, where a sequence of tokens is determined. Tokens can be words or sub-word units, depending on the task and language. These algorithms are then applied on tokens. Segmentation aims to decide if a boundary in between two tokens should be marked as a sentence (or a topic) boundary or not.

Instead of focusing on techniques used for sentence and topic segmentation individually, we first formally define these tasks and present techniques for sentence and topic segmentation in a unified framework. Then, we present the features used for segmenting text or speech.

1.1.1 Sentence Boundary Detection

Sentence boundary detection (also called *sentence segmentation*) deals with automatically segmenting a sequence of word tokens into sentence units. In written text in English and some other languages, the beginning of a sentence is usually marked with an uppercase letter, and the end of a sentence is explicitly marked with a period (.), a question mark (?), an exclamation mark (!), or another type of punctuation. However, in addition to their role as sentence boundary markers, capitalized initial letters are used to distinguish proper nouns, periods are used in abbreviations, and numbers and other punctuation marks are used inside proper names. For instance, 10% of the periods in the Brown corpus are abbreviations [24], such as "Dr." that can be an abbreviation for the words *doctor* and *drive*. And the period at the end of an abbreviation can be marking a sentence boundary at the same time, or not. For example, consider the following two sentences: *"I spoke with Dr. Smith."* and *"My house is on Mountain Dr."*. In the first sentence, the abbreviation *Dr.* does not end a sentence, and in the second it does. This percentage of periods that are used to

[1] We use the term *segmentation* to refer to both tasks.

mark an abbreviation rises to 47% in the Wall Street Journal Corpus [54]. For example, in the following sentence, partly taken from the Wall Street Journal part of the OntoNotes [37] corpus, only the last period ends the sentence:

"This year has been difficult for both Hertz and Avis", said Charles Finnie, car-rental industry analyst - yes, there is such a profession - at Alex. Brown & Sons.

Such sentences containing other sentences are not infrequent. Especially quoted sentences are always problematic, as the speaker may have uttered multiple sentences, and sentence boundaries inside the quotes are also marked with punctuation marks. An automatic method that outputs word boundaries as ending sentences according to the presence of such punctuation marks would result in cutting of some sentences wrongly. Furthermore, if the sentence above is spoken instead of written, prosodic cues usually mark structure.

Ambiguous abbreviations and capitalizations are not the only problem of sentence segmentation in written text. "Spontaneously" written texts, such as small message system (SMS) texts or instant messaging (IM) texts, tend to be nongrammatical and have poorly used or missing punctuation, which makes sentence segmentation even more challenging [98, 2].

Similarly, if the text input to be segmented into sentences comes from an automatic system, such as optical character recognition (OCR) or ASR, that aims to translate images of handwritten, typewritten, or printed text or spoken utterances into machine-editable text, the finding of sentence boundaries must deal with the errors of those systems as well. For example, [84] observed that an OCR system easily confuses periods and commas, and can result in meaningless sentences. ASR transcripts typically lack punctuation marks and are usually monocase; hence, all ASR output word boundaries can be ending or beginning a sentence. [80] asked human subjects to repunctuate monocase texts, and they performed at an F_1-measure of about 80%, which illustrates the difficulty of the task. In such input, sentence segmentation methods usually hypothesize a sentence boundary in between every two tokens.

On the other hand, for conversational speech or text, or multiparty meetings, with ungrammatical sentences and disfluencies, in most cases it is not clear where the boundary is. The inter-annotator agreement was quite low [48] during the segmentation of the Linguistic Data Consortium (LDC) distributed ICSI Meeting Corpus [41]. In an example utterance of *okay no problem*, it is not clear whether there is a single sentence or two. The problem may be redefined for the conversational domain as the task of dialog act segmentation, since dialog acts are better defined for conversational speech using a number of markup standards such as Dialog Act Markup in Several Layers (DAMSL) [16] or Meeting Recorder Dialog Act (MRDA) [76]. According to these standards, the example sentence *okay no problem* consists of two sentential units (or dialog act units): *okay* and *no problem*.

In most practical applications relying on automatic sentence segmentation, the task can be redefined according to the need of the following task. For example, the sentence "I think so but you should also ask him" may be a grammatical sentence as a whole, but for DAMSL and MRDA standards there are two dia-

log act tags, one affirmation and one suggestion. Such a modification may be needed for conversation analysis, such as speaker role detection or sentiment analysis. This task can be seen as a semantic boundary detection task instead of syntactic.

Code switching – that is, the use of words, phrases, or sentences from multiple languages by multilingual speakers – is another problem that can affect the characteristics of sentences. For example, when switching to a different language, the writer can either keep the punctuation rules from the former language or resort to the code of the later language (for instance, Spanish uses the inverted question mark to precede questions). Code switching also affects technical texts for which the meanings of punctuation signs can be redefined, as in unified resource locators (URLs), programming languages, and mathematics. One must detect and parse those specific constructs in order to process technical texts adequately.

Conventional rule-based sentence segmentation systems in well-formed texts rely on patterns to identify potential ends of sentences and lists of abbreviations for disambiguating them [54, 36, 29, 12]. For example if the word before the boundary is a known abbreviation, such as "Mr." or "Gov.", the text is not segmented at that position even though there is a period with some exceptions. While rules cover most of these cases, they do not address unknown abbreviations, abbreviations at the ends of sentences, or typos in the input text. Furthermore, such rules are not robust to text that is not well formed, such as forums, chats, or blogs, and spoken input that completely lacks typographic cues. Moreover each language requires a specific set of rules.

To improve on such a rule-based approach, sentence segmentation is stated as a classification problem. Given training data where all sentence boundaries are marked, one can train a classifier to recognize them as described in Section 1.2. Sentence segmentation in text usually uses the punctuation marks as delimiters and aims to categorize these as sentence-ending/beginning or not. On the other hand, for speech input, all word boundaries are usually considered as candidate sentence boundaries.

1.1.2 Topic Boundary Detection

Topic segmentation (sometimes referred to as *discourse or text segmentation*) is the task of automatically dividing a stream of text or speech into topically homogeneous blocks. That is, given a sequence of (written or spoken) words, the aim of topic segmentation is to find the boundaries where topics change. Figure 1.1 gives an example of a topic change boundary from a broadcast news program.

Topic segmentation is an important task for various language understanding applications, such as information extraction and retrieval, and text summarization. For example, in information retrieval, if one can segment long documents into shorter, topically coherent segments, then only the segment that is about the user's query could be retrieved.

During the late 1990s, the U.S. Defense Advanced Research Projects Agency

... Tens of thousands of people are homeless in northern China tonight after a powerful earthquake hit an earthquake registering 6.2 on the Richter scale at least 47 people are dead. Few pictures available from the region but we do know temperatures there will be very cold tonight -7 degrees. <**TOPIC_CHANGE**> Peace talks expected to resume on Monday in Belfast, Northern Ireland. ...

Figure 1.1: Example of a topic boundary in a news article.

(DARPA) initiated the Topic Detection and Tracking (TDT) program to further the state of the art in finding and following new topics in a stream of broadcast news stories [96]. One of the tasks in the TDT effort was segmenting a news stream into individual stories. While TDT established a common test bed, most researchers also use simulated environments such as by concatenating news stories from Reuters.

For multiparty meetings, the task of topic segmentation is inspired by discourse analysis. For official and well-structured meetings, the topics are segmented according to the agenda items, while for more casual conversational-style meetings, the boundaries are less clear.

Topic segmentation is a nontrivial problem without a very high human agreement because of many natural-language-related issues, and hence requires a good definition of topic categories and their granularities. For example, topics are not typically flat but occur in a semantic hierarchy. When a sentence about "soccer" is followed by a sentence about "baseball", one annotator may mark a topic change, and the other may not, considering that soccer and baseball both belong to the topic "sports". This is also the case for finer-grained distinctions. Even though the annotators are told to segment the text into a predefined number of topics, it is hard to define the concept of topic as it varies greatly, depending on the semantic content. While high inter-annotator agreement (with Cohen's kappa values of 0.7-0.9) has been achieved for the TDT corpus [19], which includes broadcast news documents and hence stories, news, or topics usually had the same boundary. For topic segmentation of multiparty meetings, the agreement is lower [38] (with kappa values of 0.6-0.7). Note that for conversational speech, the topic boundaries may not be absolute. For example, in a multiparty meeting, a few turns after switching the topic, a participant may utter a sentence about the previous topic.

In text, topic boundaries are usually marked with distinct segmentation cues, such as headlines and paragraph breaks. These cues are absent in speech. However, speech provides other cues, such as pause duration and speaker changes. This is analogous to differences between sentence segmentation of text and speech. In Section 1.5 these feature types are analyzed in more detail.

1.2 Methods

Sentence segmentation and topic segmentation have mainly been considered as a boundary classification problem. Given a boundary candidate (between two word tokens for sentence segmentation, and between two sentences for topic segmentation), the goal is to predict whether or not the candidate is an actual boundary (sentence or topic boundary) or not. Formally, let $\mathbf{x} \in \mathcal{X}$ be the vector of features (the observation) associated to a candidate, and $y \in \mathcal{Y}$ be the label predicted for that candidate. The label y can be b for boundary and $\overline{b}$ for nonboundary. This results in a classification problem: given a set of training examples $\{\mathbf{x}, y\}_{train}$, find a function that will assign the most accurately possible label y of unseen examples $\mathbf{x}_{unseen}$. Alternatively to the binary classification problem, it is also possible to model boundary types using finer-grained categories. For example, [28] suggests that sentence segmentation in text be framed as a three-class problem: sentence boundary with an abbreviation b^a, without an abbreviation $b^{\overline{a}}$, and abbreviation not at a boundary $\overline{b}^a$. Similarly, in spoken language, a three-way classification can be made between nonboundaries $\overline{b}$, statement b^s, and question boundaries b^q.

Features can be the presence of specific word n-grams around the candidate boundary, an indicator of being inside a quotation in text, an indicator of presence of the preceding word tokens in an abbreviation list, or duration of pause, pitch, energy, and other duration-related features in speech. A more detailed discussion of features is presented in Section 1.5.

For sentence or topic segmentation, the problem is defined as finding the most probable sentence or topic boundaries. The natural unit of sentence segmentation is words and of topic segmentation is sentences, as one can assume that topics typically do not change in the middle of a sentence.[2] The words or sentences are then grouped into contiguous stretches belonging to one sentence or topic, i.e., the word or sentence boundaries are classified into sentence or topic boundaries and nonboundaries. The classification can be done at each potential boundary i (local modeling); then, the aim is to estimate the most probable boundary type, $\hat{y_i}$, for each candidate example, $\mathbf{x}_i$:

$$\hat{y_i} = \underset{y_i in \mathcal{Y}}{\operatorname{argmax}} P(y_i|\mathbf{x}_i) \tag{1.1}$$

Here, the $\hat{}$ is used to denote estimated categories, and a variable without a $\hat{}$ is used to show possible categories. In this formulation, a category is assigned to each example in isolation; hence, the decision is made locally. However, the consecutive boundary types can be related to each other. For example, in broadcast news speech, two consecutive sentence boundaries, which are forming a single word sentence, are very infrequent. In local modeling, features can be extracted from the surrounding example context of the candidate boundary to model such dependencies. It is also possible to see the candidate boundaries as

[2]Similarly, it is sometimes assumed for topic-segmentation purposes that topics change only at paragraph boundaries [34].

a sequence and search for the sequence of boundary types, $\hat{Y} = \hat{y_1}, \ldots, \hat{y_n}$ that have the maximum probability given the candidate examples, $X = \mathbf{x}_1, \ldots, \mathbf{x}_n$:

$$\hat{Y} = \underset{Y}{\operatorname{argmax}} P(Y|X) \tag{1.2}$$

In the following discussion, we categorize the methods into those of local and sequence classification. Another categorization of methods is done according to the type of the machine learning algorithm: generative versus discriminative. Generative sequence models estimate the joint distribution of the observations, $P(X, Y)$, (e.g. words, punctuation) and the labels (sentence boundary, topic boundary), which requires specific assumptions (such as back-off in order to account for unseen events) and have good generalization properties. Discriminative sequence models, however, put focus on features that characterize the differences between the labeling of the examples.

Such methods (as described in the following sections) can be used for sentence and topic segmentation in both written and spoken language, with one difference: In text, the category of all boundaries that do not include a potential end-of-sentence delimiter (period, question mark, exclamation mark) is preset to nonsentence or nontopic, and a category is estimated for only those word boundaries that include a delimiter, whereas in speech, all boundaries between consecutive tokens are usually considered.

1.2.1 Generative Sequence Classification Methods

The most commonly used generative sequence classification method for topic and sentence segmentation is the hidden Markov model (HMM). The probability in Equation 1.2 is rewritten as the following, using the Bayes rule:

$$\hat{Y} = \underset{Y}{\operatorname{argmax}} P(Y|X) = \underset{Y}{\operatorname{argmax}} \frac{P(X|Y)P(Y)}{P(X)} = \underset{Y}{\operatorname{argmax}} P(X|Y)P(Y) \tag{1.3}$$

$P(X)$ in the denominator is dropped, as it is fixed for different Y, and hence does not change the argument of *max*. $P(X|Y)$ and $P(Y)$ can be estimated as

$$P(X|Y) = \prod_{i=1}^{n} P(\mathbf{x}_i|y_1, ..., y_i) \tag{1.4}$$

and

$$P(Y) = \prod_{i=1}^{n} P(y_i|y_1, ...y_{i-1}) \tag{1.5}$$

Simplifying assumptions can be made in order to make the computation of these probabilities tractable:

$$P(\mathbf{x}_i|y_1, ..., y_i) \approx P(\mathbf{x}_i|y_i) \tag{1.6}$$

and a bigram model can be assumed for modeling output categories:

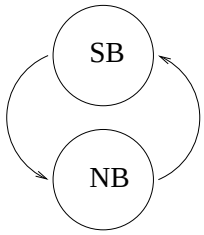

Figure 1.2: Conceptual hidden Markov model for segmentation with two states: one for segment boundaries, one for others.

Emitted Words	...	people	are	dead	few	pictures	...
State Sequence	...	NB	NB	SB	NB	NB	...

Table 1.1: Sentence segmentation with simple 2-state Markov model.

$$P(y_i|y_1, ..., y_{i-1}) \approx P(y_i|y_{i-1}) \tag{1.7}$$

The bigram case is modeled by a fully connected m-state Markov model, where m is the number of boundary categories. The states emit words (sentences or paragraphs) for sentence (topic) segmentation, and the state sequence that most likely generated the word (sentence) sequence is estimated. State transition probabilities, $P(y_i|y_{i-1})$, and state observation likelihoods, $P(\mathbf{x}_i|y_i)$, are estimated using the training data. The most probable boundary sequence is obtained by dynamic programming, thanks to the Viterbi algorithm that is used for decoding Markov models [92]. The bigram case can be extended to higher-order n-grams at the cost of an increased complexity.

For example, Figure 1.2 shows the model for the two-class problem, for example *nonboundary* (NB) and *sentence boundary* (SB) for sentence segmentation. Table 1.1 shows an example sequence of words emitted.

For topic segmentation, typically instead of using two states, n states are used, where n is the number of topics. However, obtaining state observation likelihoods without knowing the topic categories is the main challenge. [97] model topics with unigram language models (LMs) and the state observation likelihoods are trained using the k-means clustering algorithm.

Note that this is not different from using an HMM as is typically done in similar tagging tasks, such as part of speech (POS) tagging [15] or named entity extraction [7]. However, it has been shown that the conventional HMM approach has certain weaknesses. For example, it is not possible to use any information beyond words, such as POS tags of the words or prosodic cues for speech segmentation.

To this end, two simple extensions have been proposed: [77] suggested using explicit states to emit the boundary tokens, hence incorporating nonlexical information via combination with other models. This approach is used for sentence segmentation and is inspired by the hidden event language model (HELM), as

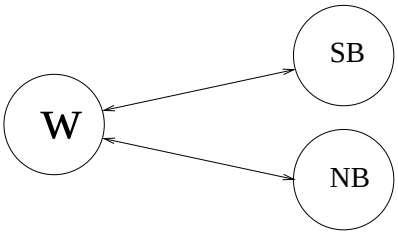

Figure 1.3: Conceptual hidden event language model for segmentation.

introduced by [81], which was originally designed for speech disfluencies. The approach was to treat such events as extra meta-tokens. In this model, one state is reserved for each boundary token, SB and NB, and the rest of the states are for generating words. To ease the computation, an imaginary token is inserted between all consecutive words, in case the word preceding the boundary is not part of a disfluency. The following example is a conceptual representation of a sequence with boundary tokens:

... people NB are NB dead YB few NB pictures ...

The most probable boundary token sequence is again obtained simply by Viterbi decoding. The conceptual HELM for segmentation is depicted in Figure 1.3.

These extra boundary tokens are then used to capture other meta-information. The most commonly used meta-information is the feedback obtained from other classifiers. Typically, the posterior probability of being in that boundary state is used as a state observation likelihood after being divided by prior probabilities [77]. These other classifiers may be trained with other feature sets as well, such as prosodic or syntactic. This hybrid approach is presented in Section 1.2.4.

For topic segmentation [89] used the very same idea and modeled topic-start and topic-final sections explicitly, which helped greatly for broadcast news topic segmentation.

The second extension is inspired from factored LMs [8], which capture not only words but also morphological, syntactic, and other information. [32] proposed using factored HELM (fHELM) for sentence segmentation using POS tags in addition to words.

1.2.2 Discriminative Local Classification Methods

Discriminative classifiers aim to model $P(y_i|\mathbf{x}_i)$ of Equation 1.1 directly. The most important distinction is that, while class densities, $p(\mathbf{x}|y)$, are model assumptions in generative approaches, such as Naive Bayes, in discriminative methods, discriminant functions of the feature space define the model. There are a number of discriminative classification approaches, such as support vector machines, boosting, maximum entropy, and regression, based on very different machine learning algorithms. While discriminative approaches have been shown to outperform generative methods in many speech and language process-

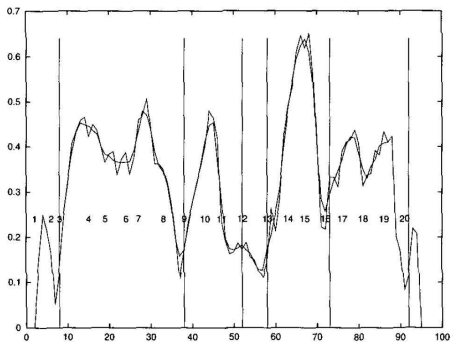

Figure 1.4: TextTiling example(from [34].

ing tasks, training typically requires iterative optimization.

In discriminative local classification, each boundary is processed separately with local and contextual features. No global (i.e., sentence or document wide) optimization is performed unlike sequence classification models. Instead, features related to a wider context may be incorporated into the feature set. For example, one can use the predicted class of the previous or next boundary in an iterative fashion.

For sentence segmentation, supervised learning methods have primarily been applied to newspaper articles. [79] used Transformation Based Learning (TBL) to infer rules for finding sentence boundaries. Many classifiers have been tried for the task: regression trees [71], neural networks [64, 40], a C4.5 classification tree [75], Maximum Entropy classifiers [68, 52], Support Vector Machines (SVMs), and Naive Bayes classifiers [28]. Mikheev treated the sentence segmentation problem as a subtask for POS tagging, by assigning a tag to punctuation similar to other tokens [59]. For tagging, he employed a combination of HMM and Maximum Entropy approaches.

The popular TextTiling method of Hearst for topic segmentation [33, 34] uses a lexical cohesion metric in a word vector space as an indicator of topic similarity. TextTiling can be seen as a local classification method with a single feature of similarity. Figure 1.4 depicts a typical graph of similarity with respect to consecutive segmentation units. The document is chopped when the similarity is below some threshold.

Originally, two methods for computing the similarity scores have been proposed: block comparison and vocabulary introduction. The first, block comparison, compares adjacent blocks of text to see how similar they are according to how many words the adjacent blocks have in common. The block size can be variable, not necessarily looking only at the consecutive blocks, but instead a window. Given two blocks, b_1 and b_2, each having k tokens (sentences or para-

graphs), the similarity (or topical cohesion) score is computed by the formula:

$$\frac{\sum_t w_{t,b_1}.w_{t,b_2}}{\sqrt{\sum_t w_{t,b_1}^2 \sum_t w_{t,b_2}^2}}$$

where $w_{t,b}$ is the weight assigned to term t in block b. The weights can be binary or may be computed using other information retrieval-based metrics such as term frequency.

The second, the vocabulary introduction method, assigns a score to a token-sequence gap based on how many new words were seen in the interval in which it is the midpoint. Similar to the above formulation, given two consecutive blocks, b_1 and b_2 of equal number of words, w, the topical cohesion score is computed with the formula:

$$\frac{NumNewTerms(b_1) + NumNewTerms(b_2)}{2 \times w}$$

where $NumNewTerms(b)$ returns the number of terms in block b, seen for the first time in text.

[11] extended this method to exploit latent semantic analysis. Instead of simply looking at all words, they worked on the transformed lexical space, which has led to improved results as this approach also captures semantic similarities implicitly.

[60] proposed using lexical chains instead of lexical similarity for estimating cohesion. Later, [44] proposed using a simpler interpretation of lexical chains, linking nonfunction words and syntactic phrases to each other only if they occur within n sentences, where n and the weights of the links are tuned based on the syntactic category.

[3] applied the original TextTiling approach to the meetings domain. [25] used a similar approach with chains of repeated terms for meeting segmentation. [38] extended this approach by using decision trees. [67] used a generative topic model with a variant of Latent Dirichlet Allocation to learn models of the topics in an unsupervised fashion, simultaneously producing a segmentation of the meetings.

[69] and [6] extended TextTiling-based methods using maximum entropy models with a wide range of lexical and discourse features tracking vocabulary shift. [26] employed SVMs for this task. [73] employed the Ripper algorithm with lexical chains, cue words, and prosodic features. [53] used decision trees with cosine similarity and prosodic features for broadcast news segmentation.

1.2.3 Discriminative Sequence Classification Methods

In segmentation tasks, the sentence or topic decision for a given example (word, sentence, paragraph) highly depends on the decision for the examples in its vicinity. Discriminative sequence classification methods are in general extensions of local discriminative models with additional decoding stages that find

the best assignment of labels by looking at neighboring decisions to label an example. Conditional Random Fields (CRFs) [51] are an extension of Maximum Entropy, SVM-Struct [88] is an extension of SVM to handle structured outputs, and maximum margin Markov networks (M^3N) are extensions of HMMs [86]. The Margin Infused Relaxed Algorithm (MIRA) is an online learning approach that requires loading of one sequence at a time during training [17]. For conciseness, we present only CRFs, which have been successful for many sequence labeling tasks, including sentence segmentation in speech.

CRFs are a class of log-linear models for labeling structures [51]. Contrary to local classifiers that predict sentence or topic boundaries independently, CRFs can oversee the whole sequence of boundary hypotheses to make their decisions. Formally, they model the conditional probability of a sequence of boundary labels ($Y = y_1, \ldots, y_n$) given the sequence of feature sets extracted from the context in which they occur ($X = \mathbf{x}_1, \ldots, \mathbf{x}_n$).

$$P(Y|X) \sim \frac{1}{Z(X)} exp\left(\sum_{t=1}^{n}\sum_{i=1}^{m}\lambda_i f_i(y_{t-1}, y_t, y_t)\right) \tag{1.8}$$
$$Z(X) = \sum_{Y} exp\left(\sum_{t=1}^{n}\sum_{i=1}^{m}\lambda_i f_i(y_{t-1}, y_t, y_t)\right)$$

where $f_i(\cdot)$ are feature functions of the observations and a clique of labels, and λ_i are the corresponding weights. $Z(\cdot)$ is a normalization function dependent only on the observations. CRFs are trained by finding the λ parameters that maximize the likelihood of the training data, usually with a regularization term to avoid overfitting. Gradient, conjugate gradient, or online methods are used for training [94, 91, 74]. Dynamic programming (Viterbi decoding) is used to find the most probable assignment of labels at test time or to compute the $Z(\cdot)$ function.

1.2.4 Hybrid Approaches

Nonsequential discriminative classification algorithms typically ignore the context, which is critical for the segmentation task. While one may add context as a feature, or simply use CRFs, which inherently consider context, these approaches are suboptimal when dealing with real-valued features, such as pause duration or pitch range. Most earlier studies simply tackled this problem by binning the feature space either manually or automatically [50].

An alternative would be to use a hybrid classification approach as suggested by [77]. The main idea would use the posterior probabilities, $P_c(y_i|\mathbf{x}_i)$, for each boundary candidate, obtained from the other classifiers, such as boosting or CRF, by simply converting them to state observation likelihoods by dividing to their priors following the well-known Bayes rule:

$$\underset{y_i}{\operatorname{argmax}} \frac{P_c(y_i|\mathbf{x}_i)}{P(y_i)} = \underset{y_i}{\operatorname{argmax}}\, P(\mathbf{x}_i|y_i) \tag{1.9}$$

Applying the Viterbi algorithm to the HMM will then return the most likely segmentation. To handle dynamic ranges of state transition probabilities and observation likelihoods, a weighting scheme as is usually described in the literature can be applied:

$$\underset{y_i}{\operatorname{argmax}} P_c(\mathbf{x}_i|y_i)^{\alpha} \times P(y_i)^{\beta} \tag{1.10}$$

where $P(y_i)$ is estimated by the HELM, α and β are optimized using a held-out set.

Zimmerman *et al.* compared various discriminative local classification methods, namely boosting, maximum entropy, and decision trees, along with their hybrid versions for sentence segmentation of multilingual speech [99]. They concluded that hybrid approaches are always superior and [32] concluded that this is also true with CRF, although to a lesser degree.

1.2.5 Extensions for Global Modeling for Sentence Segmentation

So far, most approaches to sentence segmentation have focused on recognizing boundaries rather than sentences in themselves. This has occurred because of the quadratic number of sentence hypotheses that must be assessed in comparison to the number of boundaries. To tackle that problem, [72] segment the input according to likely sentence boundaries established by a local model, and then train a reranker on the n-best lists of segmentations. This approach allows leveraging of sentence-level features such as scores from a syntactic parser or global prosodic features. [21] proposed to extend this concept to a pruned sentence lattice, which allows combining of local scores with sentence-level scores in a more efficient manner.

1.3 Complexity of the approaches

The approaches described here have advantages and disadvantages. In a given context, and under a set of observation features, one approach may be better than another. These approaches can be rated in term of complexity (time and memory) of their training and prediction algorithms, and in terms of their performance on real-world datasets. Some may also require specific preprocessing such as the conversion of continuous features to discrete features, or their normalization.

In terms of complexity, training of discriminative approaches is more complex than training of generative ones because they require multiple passes over the training data in order to adjust for their feature weights. However, generative models such as HELMs can handle multiple orders of magnitude larger training sets, and benefit, for instance, from decades of news wire transcripts. On the other hand, they work with only a few features (only words for HELM), and do not cope very well with unseen events. Discriminative classifiers allow for a wider

variety of features and perform better on smaller training sets. Predicting with discriminative classifiers is also slower, even though the models are relatively simple (linear or log-linear), because it is dominated by the cost of extracting more features.

Compared to local approaches, sequence approaches bring the additional complexity of decoding: finding the best sequence of decisions requires evaluating all possible sequences of decisions. Fortunately, conditional independence assumptions allow the use of dynamic programming to trade time for memory and decode in polynomial time. This complexity is then exponential in the order of the model (number of boundary candidates processed together) and the number of classes (number of boundary states). Discriminative sequence classifiers, such as CRFs, also need to repeatedly perform inference on the training data, which might become expensive.

1.4 Performances of the approaches

For sentence segmentation in speech, performance is usually evaluated using the error rate (ratio of number of errors to the number of examples), F_1-measure (the harmonic mean of recall and precision, where recall is defined as the ratio of the number of correctly returned sentence boundaries to the number of sentence boundaries in the reference annotations and precision is the ratio of the number of correctly returned sentence boundaries to the number of all automatically estimated sentence boundaries), and the National Institute of Standards and Technology (NIST) error rate (number of candidates wrongly labeled divided by the number of actual boundaries).

For sentence segmentation in text, researchers have reported error rate results on a subset of the Wall Street Journal Corpus of about 27,000 sentences. For instance, Mikheev [59] reports that his rule-based system performs at an error rate of 1.41%. The addition of an abbreviation list to this system lowers its error rate to 0.45%, and combining it with a supervised classifier using part-of-speech tag features leads to an error rate of 0.31%. Without requiring hand-crafted rules or an abbreviation list, Gillick's SVM-based system [28] obtains even fewer errors, at 0.25%. Even though the error rates presented seem low, sentence segmentation is one of the first processing steps for any NLP task and each error impacts subsequent steps, especially if the resulting sentences are presented to the user, as, for example, in extractive summarization.

For sentence segmentation in speech, a more commonly used evaluation measure is F_1-measure [20] report on the Mandarin TDT4 Broadcast news corpus an F_1-measure of 69.1% for a Maxent classifier, 72.6% with Adaboost and 72.7% with SVMs, using the same set of features. A combination of the three classifiers using logistic regression is also proposed. On a Turkish broadcast news corpus, [32] report an F_1-measure of 78.2% with HELM, 86.2% with fHELM with morphology features, 86.9% with Adaboost and 89.1% with CRFs. In these results, HELMs (and fHELMs) were trained on the same corpus as the other classifiers. They can, however, be trained on a much larger corpus and improve performance

when combined with discriminative classifiers. For instance, [100] reports that on the English TDT4 Broadcast news corpus, Adaboost combined with HELM performs at an F_1-measure of 67.3% compared to 65.5% for Adaboost alone.

1.5 Features

While most approaches are tightly related to the kinds of features employed, it is beneficial for demonstrative purposes to decouple these. Similarly while most feature categories are common in sentence and topic segmentation, such as lexical or prosodic features, their usage is very different. We will refer to "segmentation" when features apply to both sentence and topic segmentation, and explicitly state the kind of segmentation otherwise.

In this section, we describe the features of a potential boundary observation as the dimensions of the vector $\mathbf{x}$. A feature f can be either binary (presence of a trigger word denoted by $x_f = 1$, or absence thereof denoted by $x_f = 0$), or take values with $x_f \in \mathbb{R}$ (e.g., the length of a sentence, the duration of a pause). For binary features, in the following, we replace $x_f = 1$ by x_f and omit $x_f = 0$.

Certain classifiers assume properties for the input features, and may require that they are all binary, or may prefer that their distribution be standardized. Real-valued features can be converted to binary features by quantification and projection in a space of larger dimension, so that the value of the feature being in an interval results in its corresponding dimension in the projected space having a value of 1 while the others yield 0.

1.5.1 Features for Both Text and Speech

Lexical Features

For both text and speech, and for both sentence and topic segmentation, lexical features are the key features. Sentence and topic initial and final tokens and phrases can be captured via statistical machine learning methods as described above. Typically, windows of n tokens (or sentences) are analyzed for sentence (or topic) segmentation. While sequence classification methods perform this analysis implicitly, local classification methods can be fed corresponding features, such as the overlap of content words compared to the previous sentence.

For sentence segmentation of text, the lexical cues are tokens in text, and the task is mainly disambiguating sentence final punctuation. For speech, the lexical cues are raw tokens, as speech lacks typographic cues.

Note that lexical features have two kinds of usage. The first one is based on the occurrence of lexical features around boundaries, such as cue phrases. For example, in the Broadcast News corpus of TDT, the news, (i.e., topics) typically end with similar phrases. This first usage is described below as "discourse features". The second is similar to TextTiling-based approaches, which typically employ stems of content words that are used while computing the cosine distance. The former usage is dependent on the genre and language, while the second usage is domain independent. These two usages are not alternatives

to each other and can be combined in a single classification framework. [69] can be seen as a pioneering study for achieving this. In a maximum entropy framework, Reynar used the count of content words and names repeated in the window before and after the boundary.

More formally, let $w_1, w_2, \ldots, w_n$ be the tokens of the input, and let us extract lexical features for the boundary candidate between w_i and w_{i+1}. For sentence segmentation, the most relevant features are generally token n-grams before, after, and across the boundary. For the case of bigrams, this results in extracting the following features: x_{w_{i-1},w_i}, $x_{w_{i+1},w_{i+2}}$, and $x_{w_i,w_{i+1}}$. The cross-boundary features, for example, capture the fact that a sentence boundary is unlikely after "Gov. Smith", but likely in "government. The".

For topic segmentation, boundary candidates occur between sentences. If the sentence before the boundary is denoted s_i, and the sentence after the boundary is denoted s_{i+1}, the presence of cue phrase c in those sentences will be represented as $x_{c \in s_i}$ and $x_{c \in s_{i+1}}$. A second type of feature is the similarity of the content before and after the boundary, typically expressed as the cosine similarity between the previous and next sentences.

$$x_{cosine(s_i,s_{i+1})} = \frac{\sum_w tf(w,s_i)tf(w,s_{i+1})idf(w)}{\sqrt{\sum_w (tf(w,s_i)idf(w))^2}\sqrt{\sum_w (tf(w,s_{i+1})idf(w))^2}}$$

where $tf(w,s) = \frac{n_{w,s}}{\sum_u n_{u,s}}$ represents the term frequency of token w in sentence s and $idf(w) = \log \frac{D}{df(w)}$ is the inverse document frequency of that token, which show how common it is, generally computed on a separate corpus (D is the total number of documents, $df(w)$ is the number of documents containing w). The content can be compared at different levels, for instance n sentences before the boundary and n sentences after the boundary.

Lexical chains are another relevant feature for topic segmentation. One usually computes the number of chains that start and stop at a candidate boundary. Let $c \in \mathcal{C}$ be a set of words, referring to a lexical chain (for example, "leaf", "rose", "flower"). For practical reasons, a lexical chain is often reduced to a single token (all occurrences of "leaf"). Then, for a candidate boundary between w_i and w_{i+1}, the broken-lexical-chain feature can be computed as

$$x_{chain} = |\{c \in \mathcal{C} : \min_{\substack{w_k, w_l \in c \times c \\ k \leq i, l > i}} l - k > d_{min}\}|$$

Most automatic topic segmentation work based on text sources has explored topical word usage cues in one form or other. [49] used mutual similarity of words in a sequence of text as an indicator of text structure. [70] presented a method that finds topically similar regions in the text by graphically modeling the distribution of word repetitions. [66] extracted related word sets for topic segments with the information retrieval technique of local context analysis, and then compared the expanded word sets.

[6] combined a large set of automatically selected lexical discourse cues in a maximum-entropy model. They also incorporated topical word usage into

the model by building two statistical language models: one static (topic independent) and one that adapts its word predictions based on past words. They showed that the log likelihood ratio of the two predictors behaves as an indicator of topic boundaries, and can thus be used as an additional feature in the exponential model classifier.

Syntactic Features

Syntactic information has been successfully captured by a number of studies. Mikheev implicitly used POS tags for sentence segmentation. Similarly, for global reranking approaches as described in Section 1.2.5, syntactic features in the form of constituency trees or dependency parse trees are also used.

For morphologically rich languages, such as Czech or Turkish, morphological analyses of words are used as additional cues [32, 47].

Formally, let $t_1, \ldots, t_n$ be the sequence of part-of-speech or morphologic tags extracted for words $w_1, \ldots, w_n$. The same features can be extracted as for words (n-grams before, after, and across the candidate boundary), for example x_{t_{i-1},t_i}, $x_{t_i,t_{i+1}}$ and $x_{t_{i+1},t_{i+2}}$. Syntactic features are typically less useful for topic segmentation because topic changes are usually characterized by content shifts.

In order to assess the grammaticality of a sentence candidate in the global model under a probabilistic context free grammar (PCFG), one can compute the sum of the probability of all valid parse trees for that sentence:

$$x_{pcfg} = \sum_{t:s_i} P(t) = \sum_{t:s_i} \prod_{r \in t} P(r)$$

where t is a parse tree and r is a production rule used in that tree [42].

Discourse Features

Speech or text, discourse features are always important for segmentation. For example, in a broadcast news show, the anchor first gives the headlines, then a commercial follows, and then the stories are presented one by one with optional anchor/reporter interaction and typical topic start and end phrases.

Previous work on both text and speech segmentation has shown that cue phrases or discourse particles (items such as *now* or *by the way*), as well as other lexical cues, can provide valuable indicators of structural units in discourse [30, 65, among others]. Similarly, for speech, change of speaker may indicate a sentence boundary, and commercials may indicate a topic boundary in broadcast news or conversations. Formally, for all events $e \in \mathcal{E}$ that appear in the vicinity of a boundary, a feature x_e can be generated to represent the occurrence of that event, and if relevant, $x_{\overline{e}}$ will be used to represent the nonoccurrence of that event. Events have to be detected using additional systems not detailed in this book (such as a commercial detector) that may output confidence scores. In this case, the feature will be $x_e = cs$ where cs is the confidence score for that event to be recognized.

While earlier approaches try to capture such predetermined discourse cues, more corpus-based studies rely on the machine learning approaches to automatically learn such patterns using informative feature sets. For example, [89] used explicit HMM states for topic initial and final sentences, which improved performance greatly. [73] used statistical hypothesis testing for predetermining such phrases.

For meeting or conversation segmentation, discourse features are more complex, and rely on argumentation structure. While most studies simply use previous and next turns as discourse features, higher-level semantic information such as dialog act tags or meeting agenda items can also be used for exploiting discourse information [4].

1.5.2 Features Only for Text

Typographical and Structural Features

For sentence and topic segmentation, typographical and structural cues, such as punctuation and headlines, are very informative. Sentence segmentation systems use words and punctuation before and after the boundary, capitalization and POS tags of those words, their length, and how frequently they are used in nonsentence boundary contexts (e.g., before a lower-case word) compared to at the end/beginning of a sentence. Similarly, gazetteer information containing abbreviations and preprocessing and postprocessing patterns is employed to process text.

Formally, let g be a set of words that appear in a gazetteer, a feature is generated so that $x_{g(w)} = 1 \quad \text{iif} \quad w \in g$. Similarly, the feature that denotes the frequency of the lower case form (flc) of a word can be computed as $x_{flc(w)} = \frac{|lc(w)|}{|w|}$ where $lc(w)$ denotes the lower case version of w.

In his work on sentence segmentation, [28] observed that on a given set of features, the choice of a classifier had a much smaller impact than a mismatch between the training and the test data, or a mismatch on the tokenization of the input words. [46] proposed an unsupervised approach for finding sentence boundaries that learns abbreviations using global statistics on an unlabeled corpus. Even though the approach is independent of the language, it is unable to identify abbreviations if they are not used multiple times in the test corpus.

Other structural cues include paragraph boundaries, headlines, and section numbering. Such cues appear only in structured textual sources, and may not exist in certain text such as blogs and chatrooms.

1.5.3 Features for Speech

When working with speech recognizer output, some words may be incorrect due to recognition errors, degrading the quality of lexical features. Similarly, token start times and their durations may also be wrongly estimated, causing errors in prosodic feature computation. Typically, a large set of prosodic features are extracted for robustness to these errors.

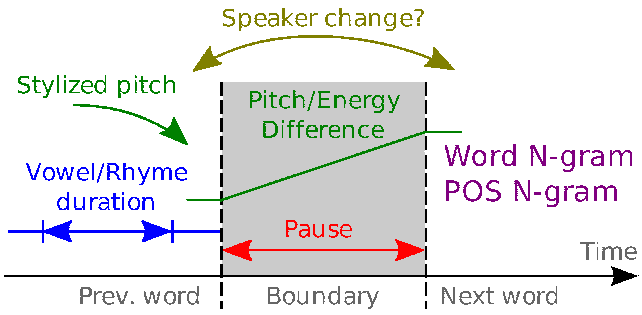

Figure 1.5: Some basic prosodic and lexical features for speech segmentation.

Prosodic Features

When applying segmentation to speech rather than written text, many of the same approaches can be used, but with some important considerations. First, in the case of automatic processing of speech, lexical information comes from the output of a speech recognizer, which typically contains errors. Second, spoken language lacks explicit punctuation, capitalization, and formatting information. Rather, this information is conveyed through the language and also through prosody, as explained below. Third, while some spoken language, for example news broadcasts, is read from a text, most natural speech is conversational. In natural spontaneous speech sentences can be "ungrammatical" (from the perspective of formal syntax) and typically contain significant numbers of normal speech disfluencies, such as filled pauses, repetitions, and repairs.

Spoken language input, on the other hand, provides additional, "beyond words" information through its intonational and rhythmic information, i.e., through its **prosody**. Prosody refers to patterns in pitch (fundamental frequency), loudness (energy), and timing (as conveyed through pausing and phonetic durations). Prosodic cues are known to be relevant to discourse structure in spontaneous speech and can therefore be expected to play a role in indicating sentence boundaries and topic transitions. Furthermore, prosodic cues by their nature are in principle independent of word identity. Thus they tend to suffer less than do lexical features from errors in automatic speech recognition.

Figure 1.5 depicts some general prosodic features used for segmenting speech into sentences, along with lexical features. Broadly speaking, the prosodic features associated with sentence boundaries are similar to those for topic boundaries, since both involve conveying a break that serves to chunk information. Pause length, and pitch and energy resets are generally greater in magnitude for the larger (i.e., topic) breaks, but similar types of prosodic features can be used for both tasks, trained of course for the task at hand.

Prosodic features for sentence segmentation have been used in a number of studies [95, 13, 77, 14, 78, 45, 53, 48, 87, 99, 55]. The simplest and most often used feature is a pause at the boundary of interest. For automatic processing, pauses are more easily obtained than other prosodic features, since unlike pitch

and energy features, pause information can be extracted from automatic speech recognition output. Of course, not all sentence boundaries contain pauses, particularly in spontaneous speech. And conversely, not all pauses correspond to sentence boundaries. For example many sentence-internal disfluencies also contain pauses. Some methods use simply the presence of a pause; others model the duration of the pause. Pause durations can be quite large in the case of turn-final sentence boundaries in conversation, since such regions correspond to time during which another participant is talking. Sentence segmentation for certain dialog acts such as backchannels (e.g., "uh-huh"), which tend to occur in isolated turns, can thus be achieved fairly successfully using only pause information.

The pause feature is computed as $x_{pause} = start(w_{i+1}) - end(w_i)$ where $start()$ and $end()$ represent the timing in seconds of the beginning and the end of a word in the speech recognition output. Relevant side features are the pause before the word (to know if it is isolated) and the quantized pause $x_{qpause}(w_i) = 1 \quad \text{iff} \quad x_{pause} > thr_{pause}$, where thr_{pause} is set to, for example, 0.2 second. Pause duration does not follow a normal distribution, by nature, and tends to confuse classifiers that expect such a distribution. However, this single feature is often the most relevant one for segmenting speech.

More detailed prosodic modeling has included pitch, phone duration, and energy information. Pitch is captured by modeling fundamental frequency during voiced regions of speech. Pitch conveys a wide range of types of information, including information about the prominence of a syllable, but for sentence segmentation the goal is usually to capture a reset in pitch. Thus, methods have looked at pitch differences across a word boundary, with a larger negative difference indicating higher probability of a sentence boundary. In addition to modeling the break in pitch across a word boundary, some approaches [77] have also modeled a speaker-specific value to which pitch falls at the ends of utterances, which not only improves performance but also allows for causal modeling, since it does not rely on speech after the pause [23].

Pitch is not a continuous function and cannot be computed outside of voiced regions. Therefore, pitch features can be undefined for a given boundary candidate, which might be a problem with certain classifiers. Computing pitch, smoothing and interpolating it properly is not the matter of this book, and should be handled by appropriate software, such as the widely used Praat toolkit [9]. Typically, features are computed from statistics of pitch values in a window before the end of the word before the candidate boundary and after the beginning of the word after the boundary. For example, the pitch difference feature described in the previous paragraph results in

$$x_{pitch} = \left(\max_{t \in W_e(w_i)} pitch(t) \right) - \left(\min_{t \in W_s(w_{i+1})} pitch(t) \right)$$

where $pitch(t)$ is the pitch value at time t, $W_e(w_i)$ is a temporal window anchored at the end of word w_i and $W_s(w_{i+1})$ is a similar window at the start of word w_{i+1}. Variants of this feature can be created by changing the window size (i.e., 200 ms, 500 ms), changing the statistics computed on both sides of

the boundary (i.e., $min, max, mean$), and normalizing pitch values according to different factors (i.e., log space projection, standardization of the distribution of pitch values of the current speaker).

Duration features for sentence segmentation aim to capture a phenomenon known as "preboundary lengthening", in which the last region of speech before the end of a unit is stretched out in duration. (Interestingly, this phenomenon is also observed in music, and even in bird song [90].) Automatic modeling methods best capture preboundary lengthening when phone durations are normalized by the average duration of those phones in a corpus of similar speaking style. The duration of the rhyme (the vowel and any following consonants) of a prefinal syllable typically shows more lengthening than does the onset of that syllable.

For example, let v be the last vowel in w_i, the word before the boundary candidate. A feature can be computed as the relative duration of that vowel compared to its average duration in a corpus $\mathcal{C}$.

$$x_{vowel} = \frac{end(v_{w_i}) - start(v_{w_i})}{\sum_{w \in \mathcal{C}} end(v_w) - start(v_w)}$$

Energy features have also been employed in sentence boundary modeling, but with less success. From a descriptive point of view, energy behaves somewhat like pitch, falling toward the end of a sentence and often showing a reset for the next sentence. However, energy is affected by a myriad of factors, including the recording itself, and can be difficult to normalize both within and across talkers. Thus it has in general been less successful than pause, pitch, and duration features for automatic segmentation.

A final feature that is sometimes considered in prosodic modeling is voice quality. Descriptive work has shown an association between sentence boundaries and voice quality changes, but because such phenomena are highly speaker dependent and difficult to capture automatically, most automatic segmentation work has relied on the previously mentioned prosodic features.

Descriptive work on topic boundaries has found that major shifts in topic typically show longer pauses, an extra-high F0 onset or "reset", a higher maximum accent peak, shifts in speaking rate, and greater range in F0 and intensity [31, 62, 35, 82, 77, among others]. Such cues are known to be salient for human listeners; in fact, subjects can perceive major discourse boundaries even if the speech itself is made unintelligible via spectral filtering [83]. In automatic studies of topic shifts, [25] found that features such as changes in speaker activity, amounts of silence and overlapping speech, and the presence of certain cue phrases were all indicative of changes in topic, and adding them to their approach improved their segmentation accuracy significantly. [27] found that similar features also gave some improvement with their approach. However, [39] found this to be true only for coarse-grained topic shifts (corresponding in many cases to changes in the activity or state of the meeting, such as introductions or closing review), and that detection of finer-grained shifts in subject matter showed no improvement.

1.6 Processing Stages

Usually, the first step in the segmentation tasks is preprocessing to determine tokens and candidate boundaries. While in language like English, words are candidate tokens, there are special cases like abbreviations and acronyms. In languages like Mandarin, with textual sources, a preceding word segmentation step can be employed.

Then a set of features, as described in the previous section, is extracted for each candidate. For speech data, token start times and durations are usually not available in the reference annotations of the spoken utterances, but these are necessary for computing prosodic features. Usually, a forced alignment of decoding step is performed to obtain these.

Once the features are extracted, each candidate boundary is classified using one of the methods described in the previous sections.

For testing, the automatically estimated token boundaries are compared to the boundaries in reference transcriptions. When speech recognizer output is used for training or testing, reference tokens are aligned with speech recognizer output words using dynamic programming to minimize alignment error (such as using NIST `sclite` alignment tools), and boundary annotations are transferred to the speech recognizer output. Unfortunately, sometimes perfect alignment is not possible. For example, if two tokens in reference annotations with a sentence boundary between them may be recognized by the speech recognizer as a single token. In such cases, it is not clear if the sentence boundary should be omitted from the speech recognizer annotations or included, so a heuristic rule is used.

1.7 Discussion

While sentence segmentation is a useful step for many language processing tasks, careful optimization of the segmentation parameters directly for the following task in comparison to independent optimization for segmentation quality of the predicted sentence boundaries has been empirically shown to be useful. For example, [93] observed that the hard-coded rules for sentence segmentation in a machine translation system resulted in very poor sentence segmentation generalization performance, compared to the use of a machine learning approach. [58] shows that optimizing parameters of sentence segmentation in the source language is useful for machine translation of spoken documents. Similarly, [22] and [56] study the effect of parameter optimization on information extraction and speech summarization, respectively, instead of optimization on the sentence segmentation task itself.

Regarding topic segmentation, automatic transcription of speech uses language models to predict topical information in the language model and this has been shown to improve ASR, either by selecting an LM trained on a matching topic or by building a general language model where the topic is a latent variable estimated during decoding. More generally, topic-driven domain adaptation is used in a wide range of natural language processing tasks. In information re-

trieval, topic is modeled explicitly [5] by allowing words to contribute differently in function of the topic in which they occur, or implicitly [18] using co-occurrence space reduction techniques. In automatic summarization, [85] proposes to reconsider the common assumption that a document is made of a single topic and includes topic-specific information in his model. Word-sense disambiguation also benefits from topic information, as many words have probably a dominant sense in a given topic [10].

1.8 Summary

We described the tasks of sentence and topic segmentation for text and speech input. We described learning algorithms for these tasks in several categories. Depending on the type of input, that is text versus speech, several different types of features may be used for these tasks. For example, while in text one can benefit from typographical cues such as capitalization and punctuation, in speech, prosodic features may be useful.

In parallel with the recent advances in speech processing and discriminative machine learning methods, performance of sentence and topic segmentation systems have improved exploiting very high dimensional feature sets. However, one must note that, these systems still make errors, requiring the follow-on processing stages, such as machine translation, to be robust to such noise. Further research is required for jointly optimizing the segmentation stage with the follow-on processing systems.

Bibliography

[1] *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, New Mexico State University, Las Cruces, NM, June 1994.

[2] A.T. Aw, M. Zhang, J. Xiao, and J. Su. A phrase-based statistical model for SMS text normalization. In *Proceedings of the COLING/ACL*, Sydney, Australia, 2006.

[3] Satanjeev Banerjee and Alexander Rudnicky. A TextTiling based approach to topic boundary detection in meetings. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Pittsburgh, PA, September 2006.

[4] Satanjeev Banerjee and Alexander Rudnicky. Segmenting meetings into agenda items by extracting implicit supervision from human note-taking. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI'07)* , Honolulu, HI, January 2007.

[5] J. Becker and D. Kuropka. Topic-based vector space model. In *Proceedings of the 6th International Conference on Business Information Systems*, pages 7–12, 2003.

[6] Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210, 1999. Special Issue on Natural Language Learning.

[7] D. M. Bikel, R. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. *Machine Learning Journal Special Issue on Natural Language Learning*, 34(1-3):211–231, 1999.

[8] J. A. Bilmes and K. Kirchhoff. Factored language models and generalized parallel backoff. In *Proceedings of the Human Language Technology Conference (HLT)-Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Edmonton, Canada, May 2003.

[9] P. Boersma and D. Weenink. Praat, a system for doing phonetics by computer, version 3.4. Technical Report 132, Institute of Phonetic Sciences of the University of Amsterdam, 1996.

[10] J. Boyd-Graber, D. Blei, and X. Zhu. A topic model for word sense disambiguation. In *Proceedings of the the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1024–1033, 2007.

[11] T. Brants, F. Chen, and I. Tsochantaridis. Topic-based document segmentation with probabilistic latent semantic analysis. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, McLean, VA, November 2002.

[12] T. Briscoe, J. Carroll, and R. Watson. The second release of the RASP system. In *Proceedings of the Interactive Demo Session of COLING/ACL*, volume 6, 2006.

[13] C. Chen. Speech recognition with automatic punctuation. In *Proceedings of EUROSPEECH*, pages 447–450, 1999.

[14] H. Christensen, Y. Gotoh, and S. Renals. Punctuation annotation using statistical prosody models. In *Proceedings of the ISCA Workshop on Prosody in Speech Recognition and Understanding*, Red Bank, NJ, USA, 2001.

[15] K. W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Conference on Applied Natural Language Processing (ANLP)*, pages 136–143, Austin, Texas, 1988.

[16] M. Core and J. Allen. Coding dialogs with the DAMSL annotation scheme. In *Proceedings of the Working Notes of the Conference of the American Association for Artificial Intelligence (AAAI) Fall Symposium on Communicative Action in Humans and Machines*, Cambridge, MA, November 1997.

[17] Koby Crammer, Ryan Mcdonald, and Fernando Pereira. Scalable large-margin online learning for structured classification. In *Annual Conference on Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2005.

[18] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[19] George Doddington. The Topic Detection and Tracking Phase 2 (TDT2) evaluation plan. In *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, February 1998. Morgan Kaufmann.

[20] M.M. Doss, D. Hakkani-Tür, O. Cetin, E. Shriberg, J. Fung, and N. Mirghafori. Entropy based classifier combination for sentence segmentation. In *Proceedings of the IEEE ICASSP Conference*, pages 189–192, 2007.

[21] B. Favre, D. Hakkani-Tür, S. Petrov, and D. Klein. Efficient sentence segmentation using syntactic features. In *Proceedings of the IEEE/ACL Spoken Language Technologies (SLT) Workshop*, Goa, India, 2008.

[22] Benoit Favre, Ralph Grishman, Dustin Hillard, Heng Ji, Dilek Hakkani-Tür, and Mari Ostendorf. Punctuating speech for information extraction. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas, NV, 2008.

[23] Luciana Ferrer, Elizabeth Shriberg, and Andreas Stolcke. Is the speaker done yet? Faster and more accurate end-of-utterance detection using prosody in human-computer dialog. In John H. L. Hansen and Bryan Pellom, editors, *Proceedings of the International Conference on Spoken Language Processing*, volume 3, pages 2061–2064, Denver, September 2002.

[24] W.N. Francis, H. Kučera, and A.W. Mackie. *Frequency analysis of English usage: Lexicon and grammar*. Houghton Mifflin, Boston MA, 1982.

[25] Michel Galley, Kathleen McKeown, Eric Fosler-Lussier, and Hongyan Jing. Discourse segmentation of multi-party conversation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2003.

[26] Maria Georgescul, Alexander Clark, and Susan Armstrong. Word distributions for thematic segmentation in a support vector machine approach. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, pages 101–108, New York, NY, June 2006.

[27] Maria Georgescul, Alexander Clark, and Susan Armstrong. Exploiting structural meeting-specific features for topic segmentation. In *Actes de la 14è me Conférence sur le Traitement Automatique des Langues Naturelles*, Toulouse, France, June 2007. Association pour le Traitement Automatique des Langues.

[28] D. Gillick. Sentence boundary detection and the problem with the U.S. In *Proceedings of NAACL: Short Papers*, 2009.

[29] G. Grefenstette and P. Tapanainen. *What is a word, What is a sentence? Problems of Tokenisation*. Rank Xerox Research Centre, 1994.

[30] B. Grosz and C. Sidner. Attention, intention, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

[31] Barbara Grosz and Julia Hirschberg. Some intonational characteristics of discourse structure. In Ohala et al. [63], pages 429–432.

[32] U. Guz, B. Favre, G. Tur, and D. Hakkani-Tür. Generative and discriminative methods using morphological information for sentence segmentation of Turkish. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(5):895–903, 2009.

[33] M. A. Hearst. Multi-paragraph segmentation of expository text. In ACL [1], pages 9–16.

[34] Marti A. Hearst. TexTiling: Segmenting text info multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.

[35] Julia Hirschberg and C. Nakatani. A prosodic analysis of discourse segments in direction-giving monologues. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 286–293, Santa Cruz, CA, June 1996.

[36] C. Hoffmann. Automatische Disambiguierung von Satzgrenzen in einem maschinenlesbaren deutschen Korpus. *Manuscript, University of Trier, Germany*, 1994.

[37] Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. OntoNotes: The 90% Solution. In *HLT-NAACL*, page 57, 2007.

[38] Pei-Yun Hsueh and Johanna Moore. Automatic topic segmentation and labeling in multiparty dialogue. In *Proceedings of the 1st IEEE/ACM Workshop on Spoken Language Technology (SLT)*, Aruba, 2006.

[39] Pei-Yun Hsueh, Johanna Moore, and Steve Renals. Automatic segmentation of multiparty dialogue. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2006.

[40] TL Humphrey and F. Zhou. Period disambiguation using a neural network. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, page 606, 1989.

[41] A. Janin, J. Ang, S. Bhagat, R. Dhillon, J. Edwards, J. Macias-Guarasa, N. Morgan, B. Peskin, E. Shriberg, A. Stolcke, C. Wooters, and B. Wrede. The ICSI meeting project: Resources and research. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Montreal, Canada, 2004.

[42] M. Johnson. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632, 1998.

[43] D. Jones, W. Shen, E. Shriberg, A. Stolcke, T. Kamm, and D. Reynolds. Two experiments comparing reading with listening for human processing of conversational telephone speech. In *Proceedings of EUROSPEECH*, pages 1145–1148, 2005.

[44] Min-Yen Kan, Judith L. Klavans, and Kathleen R. McKeown. Linear segmentation and segment significance. In *Proceedings ACL/COLING Workshop on Very Large Corpora*, Montreal, August 1998.

[45] Ji-Hwan Kim and Philip C. Woodland. A combined punctuation generation and speech recognition system and its performance enhancement using prosody. *Computer Speech and Language*, 41(4):563–577, November 2003.

[46] T. Kiss and J. Strunk. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525, 2006.

[47] J. Kolar, Y. Liu, and E. Shriberg. Genre effects on automatic sentence segmentation of speech: A comparison of broadcast news and broadcast conversations. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009.

[48] J. Kolar, E. Shriberg, and Y. Liu. Using prosody for automatic sentence segmentation of multi-party meetings. In *Proceedings of the International Conference on Text, Speech, and Dialogue (TSD)* , Czech Republic, 2006.

[49] H. Kozima. Text segmentation based on similarity between words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 286–288, Ohio State University, Columbus, OH, June 1993.

[50] H.-K. J. Kuo and Y. Gao. Maximum entropy direct models for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 14(3):873–881, 2006.

[51] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 282–289. Morgan Kaufmann Publishers Inc. San Francisco, CA, 2001.

[52] H.P. Le and T.V. Ho. A maximum entropy approach to sentence boundary detection of Vietnamesetexts. In *IEEE International Conference on Research, Innovation and Vision for the Future*, Vietnam, 2008.

[53] G. A. Levow. Assessing prosodic and text features for segmentation of Mandarinbroadcast news. In *Proceedings of the Human Language Technology Conference (HLT)-Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) 2004*, 2004.

[54] M.Y. Liberman and K.W. Church. Text analysis and word pronunciation in text-to-speech synthesis. In S. Furui and M. Mohan Sondi, editors, *Advances in Speech Signal Processing*, pages 791–831. Marcel Dekker, Inc., New York, NY, 1992.

[55] Yang Liu, Elizabeth Shriberg, Andreas Stolcke, Dustin Hillard, Mari Ostendorf, and Mary Harper. Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1526–1540, September 2006. Special Issue on Progress in Rich Transcription.

[56] Yang Liu and Shasha Xie. Impact of automatic sentence segmentation on meeting summarization. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas, NV, 2008.

[57] J. Makhoul, A. Baron, I. Bulyko, L. Nguyen, L. Ramshaw, D. Stallard, R. Schwartz, and B. Xiang. The effects of speech recognition and punctuation on information extraction performance. In *Proceedings of International Conference on Spoken Language Processing (Interspeech)*, Lisbon, Portugal, 2005.

[58] Evgeny Matusov, Dustin Hillard, Mathew Magimai-Doss, Dilek Hakkani-Tür, Mari Ostendorf, and Hermann Ney. Improving speech translation with automatic boundary prediction. In *Proceedings of International Conference on Spoken Language Processing (Interspeech)*, Antwerp, Belgium, 2007.

[59] A. Mikheev. Tagging sentence boundaries. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Athens, Greece, July 2000.

[60] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48, 1991.

[61] Joanna Mrozinski, Edward W. D. Whittaker, Pierre Chatain, and Sadaoki Furui. Automatic sentence segmentation of speech for automatic summarization. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Philadelphia, PA, 2005.

[62] S. Nakajima and J. F. Allen. A study on prosody and discourse structure in cooperative dialogues. *Phonetica*, 50:197–210, 1993.

[63] John J. Ohala, Terrance M. Nearey, Bruce L. Derwing, Megan M. Hodge, and Grace E. Wiebe, editors. *Proceedings of the International Conference on Spoken Language Processing*, Banff, Canada, October 1992.

[64] D.D. Palmer and M.A. Hearst. Adaptive sentence boundary disambiguation. In *Proceedings of the Fourth ACL Conference on Applied Natural Language Processing*, 1994.

[65] Rebecca J. Passonneau and Diane J. Litman. Discourse segmentation by human and automated means. *Computational Linguistics*, 23(1):103–139, 1997.

[66] J. M. Ponte and W. B. Croft. Text segmentation by topic. In *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries*, pages 120–129, Pisa, Italy, 1997.

[67] Matthew Purver, Konrad Körding, Thomas Griffiths, and Joshua Tenenbaum. Unsupervised topic modelling for multi-party spoken discourse. In *Proceedings of the International Conference on Computational Linguistics (COLING)-Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 17–24, Sydney, Australia, July 2006. Association for Computational Linguistics.

[68] J.C. Reynar and A. Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Conference on Applied Natural Language Processing (ANLP)*, Washington, DC, 1997.

[69] Jeffrey Reynar. Statistical models for topic segmentation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 357–364, 1999.

[70] Jeffrey C. Reynar. An automatic method of finding topic boundaries. In ACL [1], pages 331–333.

[71] Michael D. Riley. Some applications of tree-based modelling to speech and language indexing. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 339–352, 1989.

[72] B. Roark, Y. Liu, M. Harper, R. Stewart, M. Lease, M. Snover, I. Shafran, B. Dorr, J. Hale, A. Krasnyanskaya, and L. Yung. Reranking for sentence boundary detection in conversational speech. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, 2006.

[73] A. Rosenberg and J. Hirschberg. Story segmentation of broadcast news in English, Mandarin, and Arabic. In *Proceedings of the Human Language Technology Conference (HLT) and Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, New York, NY, June 2006.

[74] S. Sarawagi and W.W. Cohen. Semi-Markov conditional random fields for information extraction. *Advances in Neural Information Processing Systems*, 17:1185–1192, 2005.

[75] J. Shim, D. Kim, J. Cha, G.G. Lee, and J. Seo. Multistrategic integrated web document pre-processing for sentence and word boundary detection. *Information Processing and Management*, 38(4):509–527, 2002.

[76] E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey. The ICSI Meeting Recorder Dialog Act (MRDA) Corpus. In *Proceedings of the SigDial Workshop*, Boston, MA, May 2004.

[77] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tur. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, 32(1-2):127–154, 2000.

[78] A. Srivastava and F. Kubala. Sentence boundary detection in Arabic-speech. In *Proceedings of EUROSPEECH*, Geneva, Switzerland, 2003.

[79] E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Automatic extraction of rules for sentence boundary disambiguation. In *Proceedings of the Workshop on Machine Learning in Human Language Technology*, pages 88–92, 1999.

[80] M. Stevenson and R. Gaizauskas. Experiments on sentence boundary detection. In *Proceedings of the Conference on Applied Natural Language Processing (ANLP)*, Seattle, WA, 2000.

[81] A. Stolcke and E. Shriberg. Statistical language modeling for speech disfluencies. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Atlanta, GA, May 1996.

[82] M. Swerts. Prosodic features at discourse boundaries of different strength. *Journal of the Acoustical Society of America*, 101:514–521, 1997.

[83] M. Swerts, R. Geluykens, and J. Terken. Prosodic correlates of discourse units in spontaneous speech. In Ohala et al. [63], pages 421–424.

[84] K. Taghva, A. Condit, J. Borsack, and S. Erva. Structural markup of OCR generated text. *Information Science Research Institute 1994 Annual Research Report*, page 61, 1994.

[85] J. Tang, L. Yao, and D. Chen. Multi-topic based query-oriented summarization. In *Proceedings of SDM*, 2009.

[86] B. Taskar. *Learning structured prediction models: A large margin approach*. PhD thesis, Stanford University, 2004.

[87] M. Tomalin and P. C.Woodland. Discriminatively trained Gaussianmixture models for sentence boundary detection. In *Proceedings of ICASSP*, pages 549–552, Toulouse, France, 2006.

[88] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the International Conference on Machine Learning (ICML)*, Alberta, Canada, 2004.

[89] Gokhan Tur, Dilek Hakkani-Tür, Andreas Stolcke, and Elizabeth Shriberg. Integrating prosodic and lexical cues for automatic topic segmentation. *Computational Linguistics*, 27(1):31–57, 2001.

[90] Jacqueline Vaissière. Language-independent prosodic features. In A. Cutler and D. R. Ladd, editors, *Prosody: Models and Measurements*, chapter 5, pages 53–66. Springer, Berlin, 1983.

[91] SVN Vishwanathan, N.N. Schraudolph, M.W. Schmidt, and K.P. Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of the International Conference on Machine Learning (ICML)*, Pittsburgh, PA, 2006.

[92] A.J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, pages 1260–1269, 1967.

[93] D.J. Walker, D.E. Clements, M. Darwin, and J.W. Amtrup. Sentence boundary detection: A comparison of paradigms for improving MT quality. In *Proceedings of the MT Summit VIII*, 2001.

[94] H. Wallach. Efficient training of conditional random fields. In *Proceedings of the Annual CLUK Research Colloquium*, volume 112, 2002.

[95] V. Warnke, R. Kompe, H. Niemann, and E. Nöth. Integrated dialog act segmentation and classification using prosodic features and language models. In G. Kokkinakis, N. Fakotakis, and E. Dermatas, editors, *Proceedings of the 5th European Conference on Speech Communication and Technology*, volume 1, pages 207–210, Rhodes, Greece, September 1997.

[96] Charles L. Wayne. Topic Detection and Tracking (TDT) overview and perspective . In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, June 1998.

[97] J.P. Yamron, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt. A hidden Markov model approach to text segmentation and event tracking. In *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 333–336, Seattle, WA, May 1998.

[98] L. Zhou and D. Zhang. A heuristic approach to establishing punctuation convention in instant messaging. *IEEE Transactions on Professional Communication*, 48(4):391–400, 2005.

[99] M. Zimmerman, D. Hakkani-Tür, J. Fung, N. Mirghafori, L. Gottlieb, E. Shriberg, and Y. Liu. The ICSI+ multilingual sentence segmentation system. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Pittsburg, PA, 2006.

[100] M. Zimmerman, D. Hakkani-Tür, J. Fung, N. Mirghafori, L. Gottlieb, E. Shriberg, and Y. Liu. The ICSI+ multilingual sentence segmentation system. In *Ninth International Conference on Spoken Language Processing*. ISCA, 2006.