This article has been published in the journal "Traitement Automatique des Langues", Volume 58 (Number 2): Pages 21 to 45, Year 2017, @ATALA. The original manuscript is available on the web site <www.atala.org>.

Sentence Boundary Detection in Adjudicatory Decisions in the United States

Jaromir Savelka* — **Vern R. Walker**** — **Matthias Grabmair***** — **Kevin D. Ashley***

* *University of Pittsburgh*
** *Hofstra University*
*** *Carnegie Mellon University*

ABSTRACT. *We report results of an effort to enable computers to segment US adjudicatory decisions into sentences. We created a data set of 80 court decisions from four different domains. We show that legal decisions are more challenging for existing sentence boundary detection systems than for non-legal texts. Existing sentence boundary detection systems are based on a number of assumptions that do not hold for legal texts, hence their performance is impaired. We show that a general statistical sequence labeling model is capable of learning the definition more efficiently. We have trained a number of conditional random fields models that outperform the traditional sentence boundary detection systems when applied to adjudicatory decisions.*

RÉSUMÉ. *Nous présentons les résultats d'un effort visant à permettre aux ordinateurs de segmenter les décisions arbitrales des États-Unis en phrases. Nous avons créé un ensemble de données de 80 décisions de justice de quatre domaines différents. Nous montrons que les décisions juridiques sont plus difficiles pour les systèmes de détection des limites de peines existantes que pour les textes non juridiques. Les systèmes existants de détection des limites de phrases sont basés sur un certain nombre d'hypothèses qui ne sont pas valables pour les textes légaux, leur performance en est donc altérée. Nous montrons qu'un modèle général d'étiquetage de séquence statistique est capable d'apprendre la définition plus efficacement. Nous avons formé un certain nombre de modèles de champs aléatoires conditionnels qui surpassent les systèmes traditionnels de détection des limites de la peine lorsqu'ils sont appliqués aux décisions juridictionnelles.*

KEYWORDS: *Artificial intelligence and Law, text annotation, sentence boundary detection, conditional random fields, adjudicatory decisions.*

MOTS-CLÉS : *Intelligence artificielle et loi, annotation de texte, détection de limites de phrases, champs aléatoires conditionnels, décisions juridictionnelles.*

1. Introduction

This paper reports results of an effort to enable computers to learn to extract a particular kind of information from legal texts that human readers take for granted: segmenting the texts into sentences that express complete thoughts.

Adjudicatory decisions from the US legal system pose challenges to standard NLP techniques for sentence boundary detection (SBD). Decision makers frequently employ long sentences, complex sentence structures, quotations, citations, and extensive use of parentheses. Citations and lists introduce ambiguities in the meaning of punctuation by using periods and colons that complicate the decision of whether a sentence has ended or not. Researchers have noted that lists, with their use of colons and periods in enumerations and of citations, and their combinations of punctuation and alpha-numeric characters, make it harder to tokenize (regulatory) texts and split them into sentences (Wyner and Peters, 2011). De Maat and Winkels (2009) observed that lists degraded the performance of their sentence classifier.

SBD is a critical task in many applications such as machine translation, summarization, or information retrieval. Presumably, problems in automatically segmenting legal texts into sentences have implications for applying text processing pipelines. Errors in SBD can propagate through higher-level text processing tasks, lowering overall performance. SBD errors are particularly problematic for semantic processing of legal texts that focuses on identifying the inferential roles that sentences play, such as stating legal rules, findings of fact, or a court's conclusion of law. Suboptimal SBD will likely negatively affect the ultimate applications.

We have developed a detailed protocol setting forth guidelines for annotating sentence boundaries in legal decisions. We annotated a data set of 80 court decisions from four domains: cyber-crime decisions, intellectual property decisions, the Board of Veterans' Appeals disability decisions, and decisions of the United States Supreme Court. The complete data set contains more than 26,000 annotations (Section 4) and it is publicly available. [1]

We use the data sets to confirm that legal decisions are more challenging for existing SBD systems than the texts to which they are typically applied (i.e., news articles, short essays). We show that, if the systems are allowed to account for the peculiarities of legal decisions, their performance improves (i.e., if the systems are trained on our data sets). Most importantly, we explain that legal decisions require far more complicated definitions as to what constitutes a sentence compared to other textual data typically used in SBD work. Existing SBD systems are based on a number of assumptions that do not hold for legal text, hence their performance is impaired. We show that a general statistical sequence labeling model, such as conditional random fields (CRF), is capable of learning the definition more efficiently and can significantly outperform the traditional SBD systems in adjudicatory decisions.

1. https://github.com/jsavelka/sbd_adjudicatory_dec

2. Sentence Boundary Detection

The goal in sentence boundary detection is to split a natural language text into individual sentences (i.e., identify each sentence's boundaries). We begin with a standard definition of "sentence" from linguistic theory dealing with written linguistic structures. A sentence is a span of characters consisting of one or more words that are grammatically linked, and which is capable of expressing (at least implicitly) a complete thought. Sentences might express a declarative statement, a question, an exclamation, a request, a command, or a suggestion. A declarative sentence, for example, is an autonomous information unit that is capable of being true or false in a given situation or circumstance (Chierchia and McConnell-Ginet, 2001).

A sentence that explicitly expresses its complete thought typically contains a grammatical subject and a grammatical predicate. The grammatical subject is typically a noun phrase (a group of words that are in dependency relations with a single noun), and refers to the person, place, or thing (including abstract things) that the sentence is about. The grammatical predicate is typically a verb phrase (a group of words that are in dependency relations with a single verb, which in turn refers to an action, process or state). The predicate completes the information about the subject. An example of a normal sentence structure is "The veteran filed a claim for disability benefits," where "the veteran" is the grammatical subject and "filed a claim for disability benefits" is the grammatical predicate.

Not all sentences are as explicit in expressing their complete thoughts. An example of a one-word sentence with implicit meaning is "Yes" when it is an answer to the interrogative sentence "Did you seek medical attention for your condition?" In context, the sentence "Yes" has the same meaning as the explicit sentence "I did seek medical attention for my condition." In Section 3, we discuss other implicit sentence structures (e.g., headings and data fields).

In English, the boundary character that starts a sentence is typically an initial capital letter in the first word of the sentence (i.e., the first character within the span of characters constituting a sentence is a capital letter). Punctuation at the end of the sentence is typically the end character of the sentence span. Sentences in English typically end with one of three punctuation characters: a period (also called a "full stop"), a question mark, and an exclamation mark. In the case where a sentence is enclosed in quotation marks (either single or double), then the quotation marks are included within the sentence boundaries. Similarly, if parentheses (or other brackets) enclose a sentence, then the parentheses are included within the sentence boundaries. However, no annotation span for a sentence should start or end with a white space.

Typically, SBD is operationalized as a binary classification of a fixed number of candidate boundary points (e.g., ".", "!", "?"). For more details see Read *et al.* (2012). Approaches to SBD roughly fall into three categories:

1) *Rules* – A battery of hand-crafted matching rules is applied. The rules may look like the following:

IF "!" OR "?" MATCHED → MARK AS BOUND
(every time there is a "!" or "?" the system should consider it a boundary)
IF "<EOL><EOL>" MATCHED → MARK AS BOUND
(a boundary should be predicted every time the system encounters two consecutive line breaks)

2) *Supervised Machine Learning (ML)* – For each triggering event, decide if it is an instance of sentence boundary. Each event is represented in terms of selected features such as the following:
$x_i =<$ 0:token=".", 0:isTrigger=1, -1:token="Mr",-1:isAbbr=1, 1:token="Lange" $>$
Given the labels $y_i \in \{0, 1\}$ the supervised classifier is a function $f(x_i) \rightarrow y_i$.

Consider the following example:

> In this case, there is no question that the information Mr. Lange offered for sale was a trade secret.

The period after "Mr" is a triggering event but a system can learn that if a period follows an abbreviation then a sentence boundary should be predicted only if the following word starts with a capital letter. Unfortunately this would not work in our example. In addition the system would have to learn that "Mr" is almost always followed by a period and that it almost never ends a sentence. Therefore, the period would not be predicted as a sentence ending token.

3) *Unsupervised ML* – Similar to supervised ML approach but the system is trained on unlabeled data. The system can, for example, recognize that "Mr" is always followed by a period and therefore it is probably an abbreviation which most of the time does not end a sentence.

Multiple SBD systems were reported as having an excellent performance (Read *et al.*, 2012):

– 99.8% accuracy of a decision tree-based classifier in predicting "." as ending (or not) a sentence evaluated on the Brown corpus (Riley, 1989)

– 99.5% accuracy of a combination of an original system based on neural networks and decision trees with and existing system (Aberdeen *et al.*, 1995) evaluated on the *Wall Street Journal* corpus (WSJ) (Palmer and Hearst, 1997)

– 99.75% (WSJ) and 99.64% (Brown) accuracy of a maximum entropy model in assessing ".", "!", and "?" (Reynar and Ratnaparkhi, 1997)

– 99.69% (WSJ) and 99.8% (Brown) accuracy of a rule-based sentence splitter combined with a supervised POS-tagger (Mikheev, 2002)

– 98.35% (WSJ) and 98.98% (Brown) accuracy of an unsupervised system based on identification of abbreviations (Kiss and Strunk, 2006)

Read *et al.* (2012) conducted a study of SBD systems performance across different corpora and report more modest results ranging from 95.0% to 97.6% for different systems. Also, they tested the systems on corpora of user-generated web content. The performance of the SBD systems deteriorated for these corpora where the accuracy often falls in the lower nineties. (Read *et al.*, 2012)

3. Detecting Sentence Boundaries in US Adjudicatory Decisions

Adjudicatory decisions, whether issued by a court or by an administrative tribunal, determine whether a party's conduct conforms to the applicable legal norms, and can impose sanctions or order other remedies when the party has violated those norms. Written adjudicatory decisions are more challenging for SBD than news articles—the traditional subject of interest in developing SBD systems. Whereas news articles are generally short texts, a decision may be short but it may also be as long as a book. A decision may be structured into sections and subsections preceded by a heading (possibly numbered). A decision may contain specific constituents such as a header and a footer, footnotes, or lists. Sentences are interleaved with citations. The sentences themselves may be extremely long, or even partially spread across lists. In decisions there is a high usage of sentence organizers such as ";", or "—" and multiple types of brackets. Quotes are frequent and possibly nested.

Consider the following passage from a decision, which contains one long and complex sentence followed by a citation sentence in parentheses:

> As used in the statute, "'act in furtherance of a person's right of petition or free speech under the United States or California Constitution in connection with a public issue' includes: (1) any written or oral statement or writing made before a legislative, executive, or judicial proceeding, or any other official proceeding authorized by law; (2) any written or oral statement or writing made in connection with an issue under consideration or review by a legislative, executive, or judicial body, or any other official proceeding authorized by law; (3) any written or oral statement or writing made in a place open to the public or a public forum in connection with an issue of public interest; (4) or any other conduct in furtherance of the exercise of the constitutional right of petition or the constitutional right of free speech in connection with a public issue or an issue of public interest." (§425.16, subd. (e), italics added; see Briggs v. Eden Council for Hope & Opportunity (1999) 19 Cal. 4th 1106, 1117-1118, 1123 [81 Cal.Rptr.2d 471, 969 P.2d 564] [discussing types of statements covered by anti-SLAPP statute].)

The first sentence contains a quotation (which in turn contains a second, nested quotation) organized as a list, and it is followed by a sentence of citations and their captions. This text is very challenging for an SBD system because it spans across many triggering events that are not sentence boundaries. The second sentence is a citation and illustrates the occurrence of periods that are not sentence-ending (a common occurrence in adjudicatory decisions). The period character's common use as a sentence boundary can cause extensive segmentation errors in such decisions because it is used copiously for other purposes (e.g., abbreviations or citations).

In annotating adjudicatory texts for sentence boundaries, therefore, it is important to ensure that the annotations provide reliable and valid data. In order to ensure this, the authors adapted and used the protocol for sentence annotation developed by

the Research Laboratory for Law, Logic and Technology (LLT Lab) at the Maurice A. Deane School of Law at Hofstra University. [2] Such annotation protocols provide methods and criteria for manually annotating texts, and a set of conventions governing the generation of annotation data. Protocols are developed in two stages. First, from a sample of documents containing a variety of decisions, examples are collected that display normal forms of the annotation type, linguistic variants of those normal forms, and aberrant forms. Second, those examples are used to derive general guidelines, criteria and conventions for manually annotating these types within texts. Protocols are used not only for manually producing semantic data, but also for assuring the quality of the coding, for replicating experimental results, and for building separate but compatible datasets.

For the annotation type "Sentence", the normal form is a grammatical subject consisting of a noun phrase followed immediately by a grammatical predicate consisting of a verb phrase - i.e., <grammatical subject noun phrase><grammatical predicate verb phrase>. The noun phrase and verb phrase can contain subordinate clauses, provided the sentence as a whole is relatively easy to parse by parts of speech. In general, a span of characters is a "normal form" of an annotation type if we are highly confident that it constitutes an annotation of the specified type, and this confidence is based on some evidence or feature within the span of characters itself (an adequate "linguistic cue"). Also, a sentence in normal form has a certain fixed format or pattern, which we find recurring numerous times. Sentences having a normal form should be the easiest types of sentences for computer software to identify through standard parsing. Examples of sentences in normal form are:

> *The Veteran's chronic adjustment disorder with depressed and anxious features is related to service.*
>
> *The Veteran does meet the criteria for a diagnosis of posttraumatic stress disorder (PTSD).*
>
> *A disability which is aggravated by a service-connected disability may be service-connected.*

A span of text that is a "linguistic transform" of a normal form is one for which we are also confident that it constitutes an annotation of the type "Sentence". This confidence is based on some linguistic cue or feature within the span of text itself. However, while a sentence in normal form has a straightforward format or pattern of <grammatical subject noun phrase><grammatical predicate verb phrase>, a linguistic transform has a linguistic structure that is in principle transformable into one or more sentences that do have normal forms. There might be some linguistic rules that would make it easier for computer software to identify such forms. Examples of linguistic transforms are:

> *Consequently, as outlined in a February 2013 Formal Finding, the RO requested information from the Joint Services Records Research Center*

2. `https://github.com/jsavelka/sbd_adjudicatory_dec`

(JSRRC) and the US Army Crime Records Center; however, those sources provided negative responses for the requested date range.

Establishment of service connection for PTSD in particular requires: (1) medical evidence diagnosing PTSD; (2) credible supporting evidence that the claimed in-service stressor actually occurred; and (3) medical evidence of a link between current symptomatology and the claimed in-service stressor.

See, e.g., Young v. McDonald, 766 F.3d 1348, 1353 (Fed. Cir. 2014) ("PTSD is not the type of medical condition that lay evidence . . . is competent and sufficient to identify.").

Finally, there are spans of text that have a very particular linguistic structure (being neither normal form nor linguistic transform), but we are still confident that they constitute an instance of the type "Sentence". This confidence might be based more on the context than on linguistic cues within the span itself (e.g., co-references with words or phrases in other sentences, or standard conventions within a type of document). The following paragraphs discuss certain classes of examples on which we can generalize for adjudicatory decisions. For each class, because of its distinctive sentence structure, it would be a rather straightforward to develop more semantic information after sentence segmentation.

Case names in document titles express a single thought (one named party is suing another named party), and are best treated as a single sentence. This is true regardless of how the case name is formatted in a particular document (e.g., spread over several lines). For example, the following is a single sentence:

SOUNDEXCHANGE, INC. Plaintiff v. MUZAK, LLC Defendant.

Headings are spans of text that we annotate as "sentences" because they provide information about the organization of the text; they chunk the document into meaningful segments. For example, we understand the heading "FINDINGS OF FACT" as having a meaning similar to "The sentences in the following section state the findings of fact of the tribunal." Other standard headings in adjudicatory documents include:

INTRODUCTION

REASONS AND BASES FOR FINDINGS AND CONCLUSIONS

ORDER

Data fields are spans of text that we annotate as "sentences" because they provide the name of a data field and a value for that field in the particular document or case. They implicitly assert the value of the data field. We understand a data field such as "Decision Date: 03/28/17" as having the same meaning as "This decision was issued on March 28, 2017." Other examples are:

Citation Nr: 1710389

DOCKET NO. 12-12 279

Veteran represented by: Veterans of Foreign Wars of the United States

Page numbers of a reporter service that prints the official version of the adjudicatory decision can occur within the text file in one of two ways. First, if they occur outside of normal sentences, then we annotate them as separate sentences. For example, in the passage below, the character sequence "*1163" means "This is where page 1163 begins in the Federal Reporter, Third Series." This passage therefore contains 3 sentences, the middle one being the sentence "*1163":

> *He contends that to do so would be, in effect, to report himself for the new crime of being found in the country after deportation. *1163 See United States v. Pina-Jaime, 332 F.3d 609, 612 (9th Cir.2003) (holding that an alien need not have reentered the United States illegally to be convicted of being "found in" the country illegally).*

Second, a page number could occur embedded within a normal sentence, wherever the page break happens to fall in the printed version. In such a situation, we do not split the normal sentence into parts just because a page number happens to occur inside it. We can deal with the embedded page number in subsequent analyses, after sentence segmentation. For example, the following is segmented as a single sentence containing the page number "*1162":

> *The record in this case shows no attempt, by either the Probation Officer or sentencing court, to justify this *1162 sweeping condition.*

Ellipses (...) also occur in one of two ways. First, if the ellipsis occurs within a sentence span and it indicates missing words from within that sentence, then the ellipsis should be included within the overall sentence span. For example, the following is a single sentence that begins a block quotation, and the ellipsis occurs within the sentence boundaries:

> *... the conferee's objective was to limit the grandfather to their existing services in the same transmission medium and to any new services in a new transmission medium where only transmissions similar to their existing service are provided.*

Second, if the ellipsis occurs between sentences, then such an ellipsis should be annotated as a separate sentence. The rationale is that the ellipsis provides coded information ("Sentences have been deleted"), and should not be parsed within the complete sentences that precede or follow it. For example, the following passage contains two ellipses that we annotate as separate sentences (the first ellipsis occurs after a completed sentence and the other one occurs between paragraphs in the block quote):

> *3. The defendant shall comply with the immigration rules and regulations of the United States, and, if deported from this country, either voluntarily or involuntarily, not reenter the United States illegally. The defendant is not required to report to the Probation Office while residing out-side *1157 of the United States; however, within 72 hours of release from any custody or any reentry to the United States during the period of Court-tordered supervision, the defendant shall report for instructions to the*

United States Probation Office. . . .

. . .

5. The defendant shall not access or possess any computer or computer-related devices in any manner, or for any purpose, unless approved in advance by the Probation Officer.

Phrases functioning as complete sentences are annotated as complete sentences, as though there is an implicit ellipsis. For example, the following is only a noun phrase, but because it occurs in a list with the heading "ISSUES", we understand it to have the same meaning as the sentence "An issue in this case is whether there is entitlement to service connection for a psychiatric disorder.":

Entitlement to service connection for a psychiatric disorder.

Parentheticals within sentences occur frequently within adjudicatory decisions in the United States, especially within citations. We annotate the parenthetical as within the span of the overall sentence. This is the treatment even if, as occasionally happens, the parenthetical itself contains one or more separate sentences (i.e., the sentences within the parentheses are not annotated separately). For example, the following is a single sentence:

Id. at 576, 128 S. Ct. 558; see also id. at 575, 128 S. Ct. 558 ("The District Court began by properly calculating and considering the advisory Guidelines range. It then addressed the relevant § 3553(a) factors.").[6]

Colons as sentence-ending punctuation can sometimes occur as an exception to the normal presumption that a colon is not sentence-ending punctuation. This one exceptional situation is when the colon is the last punctuation mark in a paragraph block of text—i.e., the colon is followed immediately by a line break. A colon is therefore treated as sentence-ending punctuation if, but only if, it is followed immediately by a line break.

There are several reasons for making this exception. Although the use of a colon can be highly stylistic, in general an author uses a colon instead of a period to express that what goes before the colon is meaningfully related to what comes after – that they are in effect connected into one thought. That is why the colon is presumptively not sentence-ending punctuation. However, an author may use a colon followed immediately by a line break to introduce a block quote or an enumerated list of items. In such a situation, if we do not end the sentence with the colon, there may be no good place to end it. A block quote might contain multiple sentences or paragraphs. To include the entire block quote within the boundaries of the sentence that happens to introduce the quote would leave the block quote unsegmented. Moreover, the introductory sentence should be parsed separately from the block quote, and may have no meaning in common with the quote itself. The quoted sentence (or sentences) should be annotated independently, and parsed separately, without being part of the sentence that introduces the block quote. Similarly, a stand-alone enumerated list introduced by a colon followed by a line break should be annotated independently of the introducing sentence.

Unfortunately, a colon followed immediately by a line break might separate a grammatical subject from a list of grammatical verbs, as we sometimes find in quotations from statutory or regulatory texts. The convention we adopt here is a compromise between the desire to have first-pass segmentation that is as non-semantic as possible (comparable to tokenization) and the desire to preserve intact all propositional content in the process of sentence segmentation. We stress that this is a first-pass compromise. After this initial segmentation into sentences, on subsequent passes we can parse the spans of text before and after a <colon><line-break> to determine if they are semantically related, and if warranted we can then annotate the entire passage as (also) a single, overall sentence.

The following are two examples in which the span of the introductory sentence ends with the colon, because the colon is followed immediately by a line break:

> *For example, in a June 1977 service personnel record a counselor opined that:*
>
> *I have personally interviewed this SM and found him to have a good attitude towards the Army. However, he has a serious academic problem.*
>
> *Accordingly, the case is REMANDED for the following action:*
>
> *1. When disability ratings and effective dates have been determined for all service-connected disabilities, to include those granted herein, and all development that the RO deems necessary is undertaken, the Veteran's request for a TDIU should be readjudicated.*

Enumerated lists (whether numbered or lettered) require special treatment, and the treatment depends on whether the list items are themselves sentences or not.

If the list items are themselves sentences (including headings that we annotate as sentences), then we annotate the list number or letter itself (i.e., the number or letter of the list item) as itself a sentence, and the sentence that is the list item as another sentence. The rationale is that we would create problems for machine learning if we include the list number (e.g., "1.") as part of the sentence that is the list item. An ML program should not treat "1." as part of the sentence it introduces, or try to POS parse the sentence including the "1." within the sentence boundaries. Moreover, the "1." expresses a thought separate from the sentence it introduces. The numbering of the list could change, but the role and meaning of each sentence on the list would remain the same. For example, the following passage consists of five sentences (the heading, two list numbers, and two sentences that are list items):

> *FINDINGS OF FACT*
> *1. The Veteran does meet the criteria for a diagnosis of posttraumatic stress disorder (PTSD).*
> *2. The Veteran's chronic adjustment disorder with depressed and anxious features is related to service.*

If the list items are not themselves sentences, then there is one overall sentence that includes the list items, and the list numbers or letters occur within that overall sentence. In such a case, there is only one sentence. For example, the following is a single sentence containing an enumerated list:

> *Establishment of service connection for PTSD in particular requires: (1) medical evidence diagnosing PTSD; (2) credible supporting evidence that the claimed in-service stressor actually occurred; and (3) medical evidence of a link between current symptomatology and the claimed in-service stressor.*

Although some sentences quoted from statutes and regulations are very long and complex, we follow this same instruction in annotating them (e.g., when we annotate a block quotation of a statute or regulation that occurs within an adjudicatory decision). For example, the following consists of two sentences (the first sentence ends with the colon followed immediately by a line break):

> *These factors are:*
> *(1) the nature and circumstances of the offense and the history and characteristics of the defendant;*
> *(2) the need for the sentence imposed*
> *(A) to reflect the seriousness of the offense, to promote respect for the law, and to provide just punishment for the offense;*
> *(B) to afford adequate deterrence to criminal conduct;*
> *(C) to protect the public from further crimes of the defendant; and*
> *(D) to provide the defendant with needed educational or vocational training, medical care, or other correctional treatment in the most effective manner;*
> *(3) the kinds of sentences available;*
> *(4) the kinds of sentence and the sentencing range established for . . . the applicable category of offense committed by the applicable category of defendant as set forth in the guidelines . . .*
> *(5) any pertinent policy statement . . . issued by the Sentencing Commission . . . subject to any amendments made to such policy statement by act of Congress. . . .*
> *(6) the need to avoid unwarranted sentence disparities among defendants with similar records who have been found guilty of similar conduct; and*
> *(7) the need to provide restitution to any victims of the offense.*

Endnotes or footnotes present annotation challenges in two ways. First, the in-text indicators for endnotes or footnotes (usually numbers, but sometimes letters or other characters) should be included within the boundaries of the sentence where they occur. Sometimes they are embedded within the span of the sentence, and sometimes they occur after the sentence-ending punctuation. In the latter situation, they are still annotated as being within the span of the sentence. For example, each of the following is a single sentence (the number in square brackets being the endnote indicators):

> *Barsumyan was arrested and indicted for one count of producing, using, and trafficking in a counterfeit credit card, 18 U.S.C. § 1029(a)(1),*

> *and three counts of possession of device-making equipment, 18 U.S.C. § 1029(a)(4).[2]*
>
> *Barsumyan gave the Agent a "skimming device," [1] and asked her to covertly "skim" the hotel guests' credit cards when they registered.*

Second, if the endnotes themselves appear as a numbered list (e.g., at the end of the decision), then the annotation follows the instructions for numbered lists. The following passage would consist of four sentences (with "[4]" being the first sentence, meaning "The fourth endnote is the following.", followed by two normal sentences and a citation sentence):

> *[4] Both wireless telephones and credit cards are considered "access devices" for these purposes. The cloning of wireless telephones was considered particularly serious because cloned cell phones are commonly used by drug dealers and other criminals to evade surveillance. See 144 Cong. Rec. S3021 (1998) (statement of Sen. Leahy); 143 Cong. Rec. S2655 (1997) (statement of Sen. Kyl).*

Grammatical or typographical errors sometimes occur. Occasionally we can still determine from context and the content of the span that the span of text is a sentence (e.g., often because it is followed by a heading or a standard sentence), even if it contains grammatical or typographical errors. In such a case, we still annotate it as a sentence. For example, the following should be annotated as a sentence, despite the missing period at the end, because this span of characters was followed by a normal sentence:

> *38 U.S.C.A. § 1111 (West 2014)*

4. Data Set

We assembled a data set consisting of 80 court and administrative decisions. These came from four distinct areas of law (20 decisions from each)—appeals of veterans' disability decisions (BVA), cyber crime (CC), intellectual property (IP), and decisions of the Supreme Court of the United States (SCOTUS). We briefly describe these four data sets as well as the document selection processes in the subsections below. Selected summary statistics are provided in Table 1. The data set is publicly available. [3]

Four human annotators (the authors) marked sentence boundaries in the decisions. We were guided by the annotation protocol described earlier (Section 3). To increase the quality and consistency of the annotations we used the automatic SBD system developed by some of the authors in prior work (Savelka and Ashley, 2017). Each decision was marked by one of the annotators. First, the automatic segmenter was applied. The task of the human annotator was to correct its output.

We have double-annotated 2 randomly selected decisions from each of the areas to measure inter-annotator agreement (i.e., 8 decisions with more than 2,500 sentences).

3. `https://github.com/jsavelka/sbd_adjudicatory_dec`

		# total	longest doc	average doc	shortest doc
BVA	chars	474,478	76,255	23,723.9	9,555
(20 docs)	tokens	170,166	28,493	8,508.3	3,351
	sents	3,727	568	186.4	80
Cyber crime	chars	984,756	181,009	49,237.8	16,859
(20 docs)	tokens	367,740	71,653	18,387.0	5,986
	sents	8,295	1,613	414.8	134
IP	chars	932,133	103,974	46,606.7	15,877
(20 docs)	tokens	343,831	38,536	17,191.6	6,204
	sents	7,262	724	363.1	90
SCOTUS	chars	960,890	85,175	48,044.5	5,621
(20 docs)	tokens	355,677	31,872	17,783.9	2,130
	sents	6,768	602	338.4	62
Total	chars	3,352,257	181,009	41,903.2	5,621
(80 docs)	tokens	1,237,414	71,653	15,467.7	2,130
	sents	26,052	1,613	325.7	62

Table 1. *Summary statistics of the four data sets and their aggregate. Statistics are reported on the level of characters (chars), tokens and sentences (sents).*

The inter-annotator agreement for the different areas of law, as well as the overall agreement, is reported in Table 2. It is important to emphasize the use of the automatic SBD system in the annotation process. The agreement is most likely somewhat higher than it would be if the process was fully manual. In order to produce disagreement, one of the annotators must conclude that the automatic segmenter erred and correct its output while the other one considers it correct. If both of the human annotators correct the output, the disagreement is produced if the corrections differ.

The agreement was evaluated from two different perspectives. First, a match could be declared if:

1) *boundaries* – we count each boundary on its own; or

2) *segments* – both boundaries need to match.

Second, for each of these perspectives, two approaches could be used to determine if the boundary was predicted correctly. A match would be declared only if:

1) *strict* – boundary offsets match exactly; or

2) *lenient* – the difference between boundary offsets does not contain an alphanumeric character.

Let us consider the following example where |T| stands for the true boundary and |P| for a predicted boundary:

> |T||P|Accordingly, we find that the circuit court did not abuse its discretion when it denied Mr.|P| |P|Renfrow's motion for a JNOV.|T| |T|**|P|We find no merit to this issue.|T||P|

	BVA	Cyber Crime	IP	SCOTUS	Overall
strict-sen	.95	.93	.90	.90	.91
lenient-sen	.96	.93	.90	.91	.92
strict-bound	.97	.95	.94	.95	.95
lenient-bound	.98	.96	.95	.95	.95

Table 2. *Inter-annotator agreement.*

Two of the predicted boundaries match the true boundaries. The remaining three differ. In case of one of the three, the difference subsists in the two asterisks (non-alphanumeric). From the strict boundaries perspective (strict-bound), the Precision (P) is 0.4 and Recall (R) is 0.5. Using the lenient-boundaries perspective (lenient-bound), the P is 0.6 and R is 0.75. From the strict-segments perspective (strict-seg), both P and R are 0 (no segment is predicted correctly). Using the lenient-segments perspective (lenient-seg), the P is 0.33 and R is 0.5. Using the different perspectives allows more detailed analysis of the agreement. As shown above, a decent agreement on two boundaries does not necessarily imply that a whole segment is also predicted correctly.

Board of Veterans' Appeals Disability Decisions

We selected a sample of 20 decisions on compensation for service-related disabilities, issued by the Board of Veterans' Appeals (BVA), which is an administrative appellate tribunal within the US Department of Veterans Affairs (VA). The VA administers benefits for veterans of the US Uniformed Services, such as disability compensation, educational assistance, and other benefits. The BVA's workload has increased dramatically in the past few decades, e.g., reaching 55,713 decisions in fiscal year 2015 (Moshiashwili, 2014) (Board of Veterans' Appeals, 2015). The vast majority of appeals considered by the BVA involve claims for disability compensation (Board of Veterans' Appeals, 2015).

The decisions in this dataset are specialized to compensation for service-related disabilities, but the content of the decisions resembles the typical content of trial-level judicial decisions, with descriptions of the procedural history of the case, conclusions of law about the applicable legal rules, citations to authority and to the evidentiary record, extensive review of the evidence in the case, explanation of the tribunal's reasoning, and findings of fact on the critical legal issues. The BVA has the statutory authority to decide the facts of each case *de novo* (Moshiashwili, 2014), and it must provide a written statement of the reasons or bases for its findings and conclusions. That statement "must account for the evidence which [the BVA] finds to be persuasive or unpersuasive, analyze the credibility and probative value of all material evidence submitted by and on behalf of a claimant, and provide the reasons for its rejection of any such evidence." *Caluza v. Brown*, 7 Vet.App. 498, 506 (1995), *aff'd*, 78 F.3d 604 (Fed. Cir. 1996).

Cyber Crime, Intellectual Property, and Supreme Court Decisions

The remaining data sets comprise judicial decisions from different appellate and trial courts, times, and subject matters.

The cyber crime data set comprises of 20 decisions from criminal proceedings where the alleged offense had a strong connection to cyber space or IT technology in general. The typical offenses may involve credit card frauds, possession and distribution of electronic child pornography, or cyber bullying. The decisions were retrieved from freely accessible on-line services such as Court Listener [4] and Google Scholar [5] on the basis of hand-crafted search queries. An example query could look like this: "cybercrime unit". Because the decisions involve criminal proceedings they often emphasize fact finding and evidential reasoning.

The 20 decisions of the US Supreme Court span nearly 200 years, from the 1803 decision in Marbury v. Madison, establish the principle of judicial review, to a 2001 decision concerning a statute of limitations under the Fair Credit Reporting Act. Decisions in between those dates deal with due process and equal protection in segregation cases, the right to boycott, the WW II detention of US citizens of Japanese descent and Congress's war powers, due process and citizenship, search and seizure, the Commerce Clause and civil rights, the right to assistance of counsel under the 6th amendment, freedom of expression, birth control, presidential Executive privilege, the right to privacy, and assisted suicide. The formats of the decisions include "slip opinions," a version the Court publishes shortly after releasing a bench opinion. These may include corrections and deal with some page breaks in a complex way. We have selected some of the landmark decisions listed in the dedicated Wikipedia entry. [6]

The 20 intellectual property and related cases comprise a mix of US Supreme Court, federal Court of Appeals and federal district court cases involving issues under the federal Patent Act, 1976 Copyright Act, the Lanham Act on trademark law, and the Electronic Computer Privacy Act. Part of the data set are recent more prominent IP cases. The rest are older cases related to the IP protection of computer programs.

5. Experimental Design

We conducted a series of experiments to test several hypotheses. The first hypothesis (i.) is that court and administrative decisions are more challenging for SBD than traditional texts such as news articles. We measure performance of existing vanilla SBD systems (i.e., using the pre-trained general models) on BVA, Cyber Crime, IP, and SCOTUS data sets. The hypothesis is tested by means of comparing the measured performance with the performance of the systems reported for other types of texts (see Section 2).

4. `www.courtlistener.com`
5. `scholar.google.com`
6. `en.wikipedia.org/wiki/Lists_of_United_States_Supreme_Court_cases`

As explained in Section 3, adjudicatory decisions are subject to a number of linguistic peculiarities that make SBD a challenge. Where possible we train the existing SBD systems to explore how well they adjust to legal decisions. Specifically, we assess the hypothesis (ii.) that, by enabling the systems to account for some of the linguistic peculiarities commonly found in the decisions, their performance improves. We test the hypothesis by comparing the performance of the trained systems with that of the systems using the pre-trained general models.

There are certain assumptions about sentence boundaries that are useful for SBD on general English texts (Section 2). For example, it is useful to understand SBD in terms of binary classification of a finite set of triggering events (e.g., ".", "!", and "?") as to whether they constitute a sentence boundary or not. We test the hypothesis (iii.) that operating under these assumptions in case of decisions hurts the SBD performance by preventing the systems from accounting for some of the phenomena that regularly occur in legal texts. We implement an SBD system consisting of a general condition random fields sequence labeling model (CRF; for details see Section 6). The system does not start with any assumptions as to what could constitute a sentence. It learns the rules exclusively by means of training on labeled data. We test the hypothesis by comparing the performance of this system to the performance of the existing systems trained on our data sets.

We explore if there are peculiarities specific to different areas of law (hypothesis iv.). The point is to find out if it is more important to use decisions from the same or closely related area of law, or if it is feaible to train one general SBD system on decisions from different areas. Specifically, we train the traditional SBD models as well as the custom CRF model on one of the data sets and apply them to other data sets. We test the hypothesis by comparing the performance of the systems on the documents from the same data set to the performance on the documents from the other data sets.

For evaluation we use traditional information retrieval metrics—precision (P), recall (R), and F_1-measure (F_1). The comparison is done at the micro level, meaning that statistics are computed over all sentences across all documents in a given collection. The measures based on the alternate match criteria tend to correlate quite well. Therefore, for the sake of clarity, we only report the lenient-boundary-focused approach (one of the approaches described in Section 4).

6. Results

Vanilla SBD Systems

For evaluation of SBD systems' performance on the corpora of adjudicatory decisions we use one system from each category:

1) As an example of a system based on rules, we worked with the SBD module from the Stanford CoreNLP toolkit (Manning *et al.*, 2014). [7]

2) To test a system based on supervised ML classifier, we employed the SBD component from openNLP. [8]

3) As an example of an unsupervised system, we used the punkt (Kiss and Strunk, 2006) module from the NLTK toolkit. [9]

The criterion for selection of the SBD systems was that they are components of, we assume, widely used general NLP toolkits.

The *rule-based sentence splitter* from Stanford CoreNLP requires a text to be already segmented into tokens. The system is based on triggering events, the presence of which is a prerequisite for a boundary to be predicted. The default events are a single "." or a sequence of "?" and "!". The system may use information about paragraph boundaries which can be configured as either a single EOL (i.e., line break) or two consecutive EOLs. The system may also exploit HTML or XML markup if present. Certain patterns that may appear after a boundary are treated as parts of the preceding sentence (e.g., parenthesized expression).

The *supervised sentence splitter* from OpenNLP is based on a maximum entropy model which requires a corpus annotated with sentence boundaries. The triggering events are ".", "?", and "!". As features the system uses information about the token containing the potential boundary and about its immediate neighbors:

– the prefix

– the suffix

– the presence of particular chars in the prefix and suffix

– whether the candidate is an honorific or corporate designator

– features of the words left and right of the candidate. (Ratnaparkhi, 1998)

The *unsupervised sentence splitter* (punkt) from NLTK does not depend on any additional resources besides the corpus it is supposed to segment into sentences. The leading idea behind the system is that the chief source of wrongly predicted boundaries are periods after abbreviations. The system discovers abbreviations by testing the hypothesis $P(\cdot|w) = 0.99$ against the corpus. Additionally, token length (abbreviations are short) and the presence of internal periods are taken into account. For prediction the system uses:

– orthographic features

– a collocation heuristic (collocation is evidence against split)

– a frequent sentence starter heuristic (split after abbreviation). (Kiss and Strunk, 2006)

7. `nlp.stanford.edu/software/corenlp.shtml`
8. `opennlp.apache.org`
9. `nltk.org/api/nltk.tokenize.html`

	BVA			CC			IP			SCOTUS			Overall		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
CoreNLP	.77	.84	.81	.80	.76	.78	.77	.81	.79	.77	.76	.76	.78	.78	.78
punkt	.68	.84	.75	.72	.79	.75	.69	.80	.74	.69	.80	.74	.70	.80	.75
openNLP	.77	.81	.79	.79	.75	.77	.80	.80	.80	.77	.78	.78	.78	.78	.78

Table 3. *Vanilla SBD systems performance.*

The results of application of the three SBD systems on the four SBD data sets described in Section 4 are summarized in Table 3. The results clearly show that performance of the general SBD systems is drastically lower when compared to the performance on news articles data sets. It is also much below the reported performance on the user-generated web content. (Section 2 or Read *et al.*, 2012, for more details.) Certain portions of this gap could be explained by the particular definition of the SBD task we adopt. The remaining portion is due to the decisions being particularly challenging for SBD.

Trained SBD Systems

OpenNLP and punkt may be trained on a custom data set, which is encouraged. It can be expected that such training will improve performance of these two systems. We use the data from each data set with labeled sentence boundaries to train data-set-specialized openNLP and punkt models (^{BVA}openNLP+, ^{IP}punkt+, etc.). We also use a pooled data set (all four data set combined) to train generalized models (^{G}openNLP+ and ^{G}punkt+).

It should be noted that punkt is an unsupervised system and as such it does not use the labels in its training. Therefore, training punkt is very cheap and one could use a training set of much greater size. Indeed, we expect that if we use a larger data set to train punkt, its performance would increase beyond what we observe in our experiments. The same does not hold for openNLP, which is trained in a supervised fashion from the gold labels. Training openNLP is quite expensive. If we would wish to use more documents in training (increasing the performance further), we would have to manually label additional documents.

The CoreNLP SBD module is rule-based and therefore it is not possible to train a custom model. To approximate training, one could use its configuration options and tune the system to perform well on our data set. We configured the CoreNLP SBD module to perform well on the data sets and evaluated it alongside trained openNLP and punkt models. It should be emphasized that this kind of comparison is very problematic and it should be taken with a grain of salt. Specifically, the authors were familiar with the documents. In light of this familiarity, it appears that the CoreNLP could have an unfair advantage in this experiment.

	BVA			CC			IP			SCOTUS			Overall		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
^{BVA}punkt+	.94	.84	.89	.73	.74	.74	.68	.80	.73	.65	.80	.72	.72	.79	.75
^{CC}punkt+	.79	.86	.82	.82	.74	.78	.76	.79	.78	.71	.79	.75	.77	.79	.78
^{IP}punkt+	.80	.86	.83	.80	.74	.77	.80	.79	.80	.72	.79	.75	.78	.78	.78
^{SC}punkt+	.76	.86	.80	.79	.74	.77	.74	.80	.77	.77	.78	.78	.77	.79	.78
^{G}punkt+	.94	.85	.89	.80	.75	.77	.79	.80	.79	.75	.79	.77	.80	.79	.82
^{BVA}openNLP+	.95	.84	.89	.83	.70	.76	.83	.75	.79	.83	.73	.78	.85	.74	.79
^{CC}openNLP+	.87	.83	.85	.91	.76	.83	.90	.81	.85	.88	.78	.83	.89	.79	.84
^{IP}openNLP+	.96	.83	.89	.88	.74	.80	.93	.82	.87	.91	.77	.83	.91	.78	.84
^{SC}openNLP+	.93	.80	.86	.85	.72	.78	.88	.77	.82	.92	.79	.85	.89	.76	.82
^{G}openNLP+	.96	.84	.90	.92	.76	.83	.93	.82	.87	.92	.79	.85	.93	.80	.86
CoreNLP+	.79	.96	.87	.84	.90	.87	.80	.91	.85	.78	.83	.81	.81	.90	.85

Table 4. *Trained SBD systems performance.*

The performance of the trained (or configured) systems is summarized in Table 4. We observe that all the systems perform better when compared to the vanilla versions. The performance of some of the systems on some of the data sets is in the mid-eighties. This is still much lower than the performance reported for news articles and the performance of the models on user-generated web content (Read *et al.*, 2012).

Custom SBD Systems

We created two simple custom SBD systems. One is based on a set of handcrafted rules while the other one uses machine learning to infer the rules from our data sets. The rule-based system first replaces all sentence ending punctuation with a masking character if it occurs in an environment matching at least one of a list of manually defined regular expressions of typical legal document punctuation patterns. In a second step, the document is traversed beginning to end and begin-end-pairs are gathered for each detected sentence. It should be noted that the extracted sentences are not technically guaranteed to cover the full document. This segmenter and its set of masking regular expressions was created during a different project focusing on US Trade Secret Law decisions which had no overlapping documents with the experiments reported in this paper.

As the second system, we trained a number of conditional random fields models based on simple low-level textual features. In prior work, we showed that more complex features help to improve the performance of the system even further (Savelka and Ashley, 2017). We do not deal with this issue here and we reserve fine-tuning of the models for future work. A CRF is a random field model that is globally conditioned on an observation sequence O. The states of the model correspond to event labels E. We use a first-order CRF in our experiments (observation O_i is associated with E_i).

We use the CRFsuite [10] implementation of first-order CRF (Lafferty *et al.*, 2001; Liu *et al.*, 2005; Okazaki, 2007).

We use a very aggressive tokenization strategy that segments text into a greater number of tokens than usual. The reason for this is to capture tokens such as a single or double line breaks that may be very suggestive about sentence ending. We consider an individual token to be any consecutive sequence consisting entirely of one type of character, using the following character types:

1) letters

2) numbers

3) whitespace.

Each character that does not belong to any of the above constitutes a single token. For example, the following sequence is tokenized as shown below:

Call me at 9am on my phone (123)456-7890.

[“Call”, “ ”, “me”, “ ”, “at”, “9”, “am”, “ ”, “on”, “ ”, “my”, “phone”, “ ”, “(”, “123”, “)”, “456”, “-”, “7890”, “.”]

Each of the tokens is then a data point in a sequence that a CRF model operates on.

Each token is represented by a small set of relatively simple features. Specifically, the set includes:

1) *lower* – a token in lower case.

2) *sig* – a feature representing a signature of a token. This feature corresponds to the token with the following transformations applied:

a) each lower case letter is rewritten to “c”

b) each upper case letter is rewritten to “C”

c) each digit is rewritten to “D”.

3) *length* – a number corresponding to the length of the token in characters if the length is smaller than 4. If the length is between 4 and 6 the feature is set to “normal.” If it is greater than 6 it is set to “long.”

4) *islower* – a binary feature which is set to true if all the token characters are in lower case.

5) *isupper* – a binary feature which is set to true if all the token characters are in upper case.

6) *istitle* – a binary feature which is set to true if the first of the token characters is in upper case and the rest in lower case.

7) *isdigit* – a binary feature which is set to true if all the token characters are digits.

8) *isspace* – a binary feature which is set to true if all the token characters are whitespace.

10. `www.chokkan.org/software/crfsuite/`

	BVA			CC			IP			SCOTUS			Overall		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Custom Rules	.91	.90	.90	.74	.76	.75	.78	.82	.80	.75	.75	.75	.78	.79	.79
^{BVA}CRF	.99	.98	.99	.87	.63	.73	.87	.66	.75	.86	.65	.74	.89	.74	.77
^{CC}CRF	.96	.90	.93	.96	.92	.94	.96	.95	.96	.97	.96	.96	.95	.95	.95
^{IP}CRF	.97	.90	.93	.96	.90	.93	.97	.95	.96	.97	.94	.95	.96	.94	.95
^{SC}CRF	.95	.87	.91	.94	.90	.92	.93	.93	.93	.97	.94	.95	.95	.93	.94
^{G}CRF	.99	.99	.99	.95	.94	.95	.95	.96	.95	.97	.96	.96	.97	.95	.96

Table 5. *Custom SBD systems performance.*

In addition, for each token we also include *lower*, *sig*, *islower*, *isupper*, *istitle*, *isdigit*, and *isspace* features from the three preceding tokens and three following tokens. If one of these tokens falls beyond the document boundaries, we signal this by including *BOS* (beginning of sequence) and *EOS* (end of sequence) features.

Taking a look at the "Call me at 9am ..." sequence from the above example, the third token of this sequence ("me") would be represented along the following lines:

```
{bias, 0:lower=me, 0:sig=cc, 0:length=2, 0:islower=true,
 0:isupper=false, 0:istitle=false, 0:isdigit=false,
 0:isspace=false, -3:BOS, -2:lower=call, -2:sig=Cccc,
 -2:length=normal, -2:islower=false, -2:isupper=false,
 -2:istitle=true, -2:isdigit=false, -2:isspace=false,
 -1:lower=" ", -1:sig=" ", -1:length=1, -1:islower=false,
 -1:isupper=false, -1:istitle=false, -1:isdigit=false,
 -1:isspace=true, 1:lower=" ", 1:sig=" ", 1:length=1,
 1:islower=false, 1:isupper=false, 1:istitle=false,
 1:isdigit=false, 1:isspace=true, 2:lower=at, 2:sig=cc,
 2:length=2, 2:islower=true, 2:isupper=false, 2:istitle=false,
 2:isdigit=false, 2:isspace=false, 3:lower=" ", 3:sig=" ",
 3:length=1, 3:islower=false, 3:isupper=false, 3:istitle=false,
 3:isdigit=false, 3:isspace=true}
```

As labels we use the Sentence annotation type projected into the BILOU [11] as demonstrated on the following example:

> Look! It is here.
>
> ["Look", "!, " ", "It", " ", "is", " ", "here", "."]
>
> [B-Sentence, L-Sentence, O, B-Sentence, I-Sentence, I-Sentence, I-Sentence, I-Sentence, L-Sentence]

11. B: beginning of sequence, I: inside sequence, L: last in sequence, O: outside of sequence, U: unit-length sequence.

The performance of the custom SBD systems is reported in Table 5. For most of the data sets the performance of some of the models reaches the middle nineties. This performance is comparable to the performance of the traditional SBD models on user-generated web content (Read *et al.*, 2012). For a small number of data sets, the performance is in the higher nineties comparable to the performance of SBD systems on news articles (Read *et al.*, 2012).

7. Discussion

The results of applying off-the-shelf SBD systems on legal decisions (Table 3) clearly show that the performance of the general SBD systems is drastically lower as compared to their performance on news articles data sets. It is also much below the reported performance on the user generated web content (Read *et al.*, 2012). Certain portions of this gap could be explained by the particular definition of the SBD task we adopt (Section 3). The remaining portion is due to the decisions being particularly challenging for SBD.

The most common source of errors is due to wrongly predicted sentence boundaries in citations as shown in the example:

> see United States v. X-Citement Video, Inc., 513 U.S. 64, 76-78, 115 S. Ct. 464, 130 L. Ed.|P| 2d 372 (1994)

The predicted boundary is marked with |P|. This type of error is very serious because it causes broken sentences to be passed along for further processing within the pipeline. These sentences may eventually even show up in the output presented to a user (e.g., in a summary).

Another commonly occurring type of error is a missed boundary that follows a unit if a triggering event is absent:

> 1)|T| Response to Jury Question|T|

The true boundary is marked with |T|; the absence of a predicted boundary |P| indicates an error. This type of error is partly caused by our specific definition of SBD. This type of mistake is less serious than the previous one. It may still negatively affect the performance of the processing pipeline but it does not introduce broken sentences that may eventually be output to a user.

The performance of the trained (or configured) systems universally improved over their off-the-shelf counterparts (compare Table 4 and Table 3). Even though the performance of CoreNLP improved, the wrongly predicted boundaries in citations remain a problem. Below are two examples of the boundaries that were incorrectly predicted by CoreNLP+:

1) Entick v. Carrington, 95 Eng.|P| Rep. 807 (C. P. 1765)

2) 451 F. Supp.|P| 2d 71, 88 (2006).

The training improved the performance of the general openNLP SBD module dramatically when it comes to precision. The performance in terms of recall remained about the same. Although some of the boundaries are missed, it is quite rare for openNLP+ to predict an incorrect boundary. Systematic errors are mostly missed boundaries such as those in the following examples:

1) 5. The Government's Hybrid Theory|T|

2) This device delivers many different types of communication: live conversations, voice mail, pages, text messages, e-mail, alarms, internet, video, photos, dialing, signaling, etc.|T| The legal standard for government access depends entirely upon the type of communication involved.

In the first example the system missed a boundary because it is not associated with a triggering event (heading). Example 2 is interesting because the system obviously learned that the "etc." is an abbreviation which often does not end a sentence.

The trained punkt+ performs better than the general one. It still commits slightly more errors as compared to the other two trained/configured systems. One would probably need to train punkt+ on a considerably larger data set in order to match the performance of the other two systems. The previously identified typical errors occur:

1) II. ANALYSIS|T|

2) "[T]he district court retains broad discretion in deciding how to respond to a question propounded from the jury and . . . |P| the court has an obligation to dispel any confusion quickly and with concrete accuracy."

Example 1 shows a missed boundary after a heading. Example 2 shows a wrongly predicted boundary after three dots in a quotation.

The custom CRF system clearly outperforms both vanilla and trained general systems on all four data sets, suggesting that our general hypothesis holds. Although the system performs quite well, there is certainly room for improvement. Here are some examples of errors in predicting boundaries:

1) Such a procedure, this Court said, "cannot be an adequate substitute for the right to full appellate review available to all defendants"|P| *743 who may not be able to afford such an expense.

2) *654|T| III.

3) "The introduction of this article declares the opinion.|P| . . that Congress could not declare"

4) "It settles the great question of citizenship and removes all doubt as to what persons are or are not citizens of the United States.|T| . . . We desired to put this question of citizenship and the rights of citizens .|P| . . under the civil rights bill beyond the legislative power . . . ."

Both examples 1 and 2 relate to the phenomenon of editorial content inserted into the text of a decision. These are page numbers indicating that there is a page break in

a printed document. Dealing with this phenomenon is difficult because we treat these as standalone sentences if they fall in between two other sentences but we ignore them if they are embedded within a single sentence. Examples 3 and 4 show mishandling of ellipses. Dealing with ellipses is difficult and they are a source of many errors.

8. Future Work

For future work we would like to use the data set that we have assembled to train more powerful sentence boundary detectors. Despite the nice improvement over the traditional SBD systems, the number of errors is still considerable. The prediction model that we used here is quite simple (a single CRF model). We have already shown that chaining multiple models together improves the SBD performance (Savelka and Ashley, 2017). There the focus was on distinguishing the main and the auxiliary content first and then using this information in decisions about sentence boundaries. A similar setup could probably lead to even better results than those reported in Savelka and Ashley (2017) because we now have a significantly richer and larger data set. In addition, more recent (and presumably more effective) sequence labeling models than CRFs could be employed (e.g., long short-term memory networks).

9. Conclusion

We assembled a data set consisting of 80 court and administrative decisions. These came from four distinct areas of law. We annotated the decisions with sentence boundaries producing a data set that consists of more than 24,000 sentences. We used the data set to show that court and administrative decisions are more challenging for SBD than traditional texts such as news articles. This is due to some peculiar linguistic features that regularly occur in adjudicatory decisions. We confirmed this by training the available SBD systems on our data set observing a visible improvement in performance. We also explored the usefulness of typical assumptions traditional SBD systems operate on. We found that operating under these assumptions for legal decisions hurts the SBD performance. It prevents the systems from accounting for some of the phenomena that regularly occur in legal texts. A general CRF model trained on our data set performed significantly better than the traditional SBD systems.

Acknowledgements

This work was supported in part by the National Institute of Justice Graduate Student Fellowship (Fellow: Jaromir Savelka) Award # 2016-R2-CX-0010, "Recommendation System for Statutory Interpretation in Cybercrime." This work was also supported in part by the Maurice A. Deane School of Law at Hofstra University, through its general support of the Research Laboratory for Law, Logic and Technology and through its support of this particular research project.

10. References

Aberdeen J., Burger J., Day D., Hirschman L., Robinson P., Vilain M., "MITRE: description of the Alembic system used for MUC-6", *Proceedings of the 6th Conference on Message Understanding*, Association for Computational Linguistics, p. 141-155, 1995.

Board of Veterans' Appeals U. S. D. o. V. A., "Annual Report: Fiscal Year 2015", 2015.

Chierchia G., McConnell-Ginet S., *Meaning and grammar: An introduction to semantics*, Second Edition, MIT press, 2001.

de Maat E., Winkels R., "A next step towards automated modelling of sources of law", *Proceedings of the 12th International Conference on AI and Law*, ACM, p. 31-39, 2009.

Kiss T., Strunk J., "Unsupervised multilingual sentence boundary detection", *Computational Linguistics*, vol. 32, nº 4, p. 485-525, 2006.

Lafferty J., McCallum A., Pereira F. *et al.*, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", *Proceedings of the 18th International Conference on Machine Learning, ICML*, vol. 1, p. 282-289, 2001.

Liu Y., Stolcke A., Shriberg E., Harper M., "Using conditional random fields for sentence boundary detection in speech", *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, p. 451-458, 2005.

Manning C. D., Surdeanu M., Bauer J., Finkel J. R., Bethard S., McClosky D., "The stanford corenlp natural language processing toolkit", *ACL*, p. 55-60, 2014.

Mikheev A., "Periods, capitalized words, etc.", *Computational Linguistics*, vol. 28, nº 3, p. 289-318, 2002.

Moshiashwili V. H., "The Downfall of Auer Deference: Veterans Law at the Federal Circuit in 2014", *Am. UL Rev.*, vol. 64, p. 1007, 2014.

Okazaki N., "CRFsuite: a fast implementation of Conditional Random Fields", 2007.

Palmer D. D., Hearst M. A., "Adaptive multilingual sentence boundary disambiguation", *Computational Linguistics*, vol. 23, nº 2, p. 241-267, 1997.

Ratnaparkhi A., Maximum entropy models for natural language ambiguity resolution, PhD thesis, University of Pennsylvania, 1998.

Read J., Dridan R., Oepen S., Solberg L. J., "Sentence boundary detection: A long solved problem?", *COLING (Posters)*, vol. 12, p. 985-994, 2012.

Reynar J. C., Ratnaparkhi A., "A maximum entropy approach to identifying sentence boundaries", *Proceedings of the 5th Conference on Applied Natural Language Processing*, Association for Computational Linguistics, p. 16-19, 1997.

Riley M. D., "Some applications of tree-based modelling to speech and language", *Proceedings of the Workshop on Speech and Natural Language*, ACL, p. 339-352, 1989.

Savelka J., Ashley K. D., "Using Conditional Random Fields to Detect Different Functional Types of Content in Decisions of United States Courts with Example Application to Sentence Boundary Detection", *Proceedings of the 2nd Workshop on Automated Detection, Extraction and Analysis of Semantic Information in Legal Texts*, ICAIL, 2017.

Wyner A., Peters W., "On Rule Extraction from Regulations.", *JURIX*, vol. 11, p. 113-122, 2011.