Sentence Boundary Detection in Legal Text

George Sanchez
Thomson Reuters R&D
610 Opperman Dr. Eagan, MN 55123
FName.LName@TR.com

Abstract

In this paper, we examined several algorithms to detect sentence boundaries in legal text. Legal text presents challenges for sentence tokenizers because of the variety of punctuations, linguistic structure, and syntax of legal text. Out-of-the-box algorithms perform poorly on legal text affecting further analysis of the text. A novel and domain-specific approach is needed to detect sentence boundaries to further analyze legal text. We present the results of our investigation in this paper.

1 Introduction

Sentence Boundary Detection (SBD) is an important fundamental task in any Natural Language Processing (NLP) application because errors tend to propagate to high-level tasks and because the obviousness of SBD errors can lead users to question the correctness and value of an entire product. While SBD is regarded as a solved problem in many domains, legal text presents unique challenges. The remainder of this paper describes those challenges and evaluates three approaches to the task, including a modification to a commonly-used semi-supervised and rule-based library as well as two supervised sequence labeling approaches. We find that a fully-supervised approach is superior to the semi-supervised rule library.

2 Previous Work

There are several out-of-the-box algorithms for SBD. Most of these algorithms are available in the most commonly used natural language processing (NLP) libraries. These algorithms for SBD are a product of years of research and study of natural language processing. SBD has not recently received much attention from the NLP community (Read, 2012) and is almost always considered a side issue in most NLP research efforts (Walker, 2001). Most of the algorithms like decision tree classifier (Riley, 1989), Naïve Bayes and Support Vector Machine (SVM) based models as reviewed in (Gillick, 2009), and the Punkt unsupervised model (Kiss, 2006) proved to be highly accurate and adequate for most domain language data, such as collections of news articles. These algorithms common in NLP toolkits often perform rather poorly in specific domains like the biomedical domain (Griffis, 2016). We observe the same poor performance on legal and tax documents when an untrained unmodified PunktSentenceTokenizer in NLTK (Bird, 2009) was used in Section 5. Algorithms such as Punkt, need to be customized and trained for a specific domain to be effective.

3 Experiments

We reviewed the following approach:

- Punkt Model with Custom Abbreviations
- Conditional Random Field
- Deep Learning Neural Networks

In-house rule-based SBD gathered from subject matter experts in the organization, where the author is employed at, is reaching its limits when used to process newer legal documents because more current legal documents have complex structures. Python's Natural Language Toolkit (NLTK) Punkt model proved to be good enough for some time that we were able to develop several customized models. This unsupervised model approach with Punkt allowed us greater flexibility to adapt to several collections of legal and tax documents.

Proceedings of the Natural Legal Language Processing Workshop 2019, pages 31–38
Minneapolis, Minesotta, June 7, 2019.

Nonetheless, we still found some situations that the customized Punkt models were not able to handle. So, we started looking into other methods for SBD. In this paper, we compare Punkt, our customized Punkt model, a Conditional Random Field (CRF) model, and a deep learning neural network approach. Our investigation used the publicly available data in (Savelka, 2017), available at https://github.com/jsavelka/sbd_adjudicatory_dec/tree/master/data_set

4 SBD Challenges in Legal Text

Legal texts are more challenging for SDB than most domain language data like news articles because most legal texts are structured into sections and subsections, not like the narrative structure of a news article. Legal texts contain some or all of the following elements header, footer, footnotes, or lists. The most important sections or discussion in legal text are interleaved with citations. The sentences may be extended or spread across lists.

Legal texts also have a very particular linguistic structure. These are spans of text that doesn't follow a standard sentence structure but is considered important part of the text. Here are some problematic language structures that affect SBD in legal text (Savelka, 2017):

- **Case names in document** titles are best treated as a sentence legal text. e.g., *UNITED STATES of America, Plaintiff–Appellee, v. Matthew R. LANGE, Defendant–Appellant.*

- **Headings** in legal text provides information about the organization of the text. The headings chunk the text into meaningful segments or issues. e.g., ARGUMENT

  INTRODUCTION

  I. BACKGROUND

- **Fields** with values that provide the name of the field e.g.,

  DOCKET NO. A–4462–13T2

  Prison Number: #176948

- **Page Numbers** that refers to reporter service prints containing cited or discussed text e.g., *59 During a search of the defendant's closed, but unlocked

  *1163 See United States v. Pina-Jaime, 332 F.3d 609, 612 (9th Cir.2003)

- **Ellipses** in sentences indicate missing words or indicate that some sentences have been deleted. This is often used for quoted text. e.g., *...After granting discretionary review, the Supreme Court, Aker, J., held that rule, which stated that court*

- **Parentheticals** within sentences often occur with citations. e.g., see also United States v. Infante-Ruiz, *13 F.3d 498, 504-505 (1st Cir.1994)* (when third party consent to search vehicle and trunk is qualified by a warning that the briefcase belonged to another, officers could not assume without further inquiry that the consent extended to the briefcase)

- **Enumerated lists** (whether numbered or lettered) e.g.,

  FINDINGS OF FACT

  1. The Veteran does meet the criteria for a diagnosis of posttraumatic stress disorder (PTSD).

- **Endnotes or footnotes** text indicator often occur near sentence boundaries. e.g., *and three counts of possession of device-making equipment, 18 U.S.C. § 1029(a)(4).[2]*

- **Citations in sentences** e.g., Thus, even an “infinitesimal contribution to the disability might require full contribution.” *(Id., at pp. 430–431, 133 Cal.Rptr. 809.)* The Heaton court also rejected this argument, noting that section 31722 explicitly provided for mental as well as physical disabilities.

The dataset used for the experiments in this paper were carefully annotated to address how each of the described situations above was treated. Protocols were put in place to have consistency as to what is considered a "sentence" in the dataset.

Please see (Savelka, 2017) for a thorough discussion of how the sentences in the dataset were annotated.

5 Data

The dataset contains decisions in the United States Courts (Savelka, 2017). The dataset is in four files bva.json, cyber_crime.json, intellectual_property.json, and scotus.json. All of the experiments used the bva.json, intellectual_property.json, and scotus.json for training and development of the model. Each model was tested on cyber_crime.json. Each file contains several decisions with the full text of the decision and a list of offsets of sentence boundaries in the text. The sentences were extracted using the offsets provided to prepare the data for training. Each sentence was tokenized to create breaks between numbers, alphabetic characters, and punctuation. Then, each token was labeled 'B' – Beginning 'I' -Inside, 'L'–Last. Example:

Sentence:

```
See United States v. Bailey, 227
F.3d 792, 797 (7th Cir.2000);
```

Tokens:

```
['See', 'United', 'States', 'v',
'.', 'Bailey', ',', '227', 'F', '.
', '3', 'd', '792', ',', '797', '
(', '7', 'th', 'Cir', '.', '2000',
')', ';']
```

Label:

```
['B', 'I', 'I', 'I', 'I', 'I', 'I
', 'I', 'I', 'I', 'I', 'I', 'I', '
I', 'I', 'I', 'I', 'I', 'I', 'I',
'I', 'I', 'L']
```

After pre-processing all the Adjudicat ory decisions, the dataset contains 80 documents, 26052 sentences, and the following distribution of labels.

```
{'B': 25126,'I': 658870, 'L': 26052}
```

6 Punkt Experiment

The Punkt model is an unsupervised algorithm with an assumption that SBD can be improved if abbreviations are correctly detected and then eliminated (Kiss, 2006). In our investigation of Punkt, we used the PunktSentenceTokenizer without further training and a trained instance with modified abbreviations.

6.1 Punkt (unmodified/untrained) Results

Predicted	Actual		
	B	I	L
B	6652	1019	103
I	3121	188114	3525
L	136	1298	6861

Table 1– Punkt(untrained/unmodified) Confusion Matrix

	Precision	Recall	F1-Score	Support
B	0.671	0.856	0.752	7774
I	0.988	0.966	0.977	194760
L	0.654	0.827	0.731	8295
Weighted Avg	0.963	0.956	0.959	210829
Weighted Avg Excluding 'I'	0.662	0.841	0.741	16069

Table 2 – Punkt (untrained/unmodified) Results Report

Table 2 gives us a summary of the results for Precision, Recall, F1-score, and Support of the experiment. The support column is the number of elements in each class label. The majority of labeled tokens are "I" (Inside), but the end of sentence labels "L" (Last), which we need to target, are most important because it is the token label that represents the end of sentences. The unmodified and untrained PunktSentenceTokenizer gave us a weighted average F1-Score of 0.959 (Table 2) including the predictions for tokens "I" (Inside) the sentences. Excluding "I" (Inside), we get a weighted average F1-Score of only 0.741. A precision of around 65.4% (precision for "L" (Last) on Table 2) detecting end of sentences might not be adequate when processing large volumes of legal text. In comparison, (Kiss, 2006) reports an error rate of 1.02% (98.98% Precision) [2] on the Brown corpus and 1.65% (98.35% Precision) on the WSJ. To improve the score of the Punkt model, we trained it and gave it an updated abbreviation list based on the legal text domain. Please see (Appendix A 1) for a sample list of abbreviations that we included in the model.

Additionally, we replace '\n'(newline) and '\t' (tab) with space and "(double quotes) with '' (two single quotes) for each sentence used for training. The model was trained to learn parameters unsupervised using the cleaned sentences of training set files. To test the Punkt model, we used the test set file. The test labels

2 Precision calculated as 100 – error rate

were generated using the B, I, and L labels as described in (Section 4). Each document in the test set was sentence tokenized using the model then assigned the appropriate B, I, and L labels to generate the predicted labels for the test file.

6.2 Punkt (trained/updated) Results

Predicted	Actual		
	B	I	L
B	6640	1031	103
I	2258	189844	2658
L	135	1311	6849

Table 3 – Punkt (trained/updated) Results Confusion Matrix

	Precision	Recall	F1-Score	Support
B	0.735	0.854	0.790	7774
I	0.988	0.975	0.981	194760
L	0.713	0.826	0.765	8295
Weighted Avg	0.968	0.964	0.966	210829
Weighted Avg Excluding 'I'	0.724	0.839	0.777	16069

Table 4 – Punkt (trained/updated) Results Report

For Punkt, the added abbreviation makes the model slightly better than without training and adding the abbreviation. For the trained and updated model, we get a weighted average F1-Score of 0.966 including the "I" (Inside) labels and 0.777 excluding the "I" (Inside). We see about the same F1-score on the weighted average excluding the "I" (Inside) labels. The precision on "L" (Last) labels slightly increased as well. Precision is still low compared to the performance of Punkt against the Brown corpus and WSJ (Kiss, 2006).

6.3 Punkt SBD Errors

Both Punkt(untrained/unmodified) and Punkt(trained/updated) used periods to segment sentences which do not work well with legal text.

Sentence segmentation as labeled:

```
In Re Application for Pen Register and Trap/Trace, 396 F. Supp. 2d 747 (S.D. Tex. 2005)
District Court, S.D. Texas
Filed: October 14th, 2005
```

Punkt segmentation (untrained/unmodified and trained/updated):

```
In Re Application for Pen Register and Trap/Trace, 396 F. Supp.
2d 747 (S.D.
Tex.
2005) District Court, S.D.
Texas Filed: October 14th, 2005
```

7 Conditional Random Field Experiment

After Punkt, we evaluate the use of Conditional Random Field (CRF) for SBD. A CRF is a random field conditioned on an observation sequence (Liu, 2005). A sentence is an excellent example of an observation sequence. CRF's are being used successfully for a variety of text processing tasks (Liu, 2005). We build on Savelka's (2017) work on using CRF for SDB for legal text.

7.1 Feature Extraction for CRF Experiment

Features were extracted for each token using a window of 3 tokens before and after the token that is in focus. For those 3 tokens before and after, we extracted a total of 8 features based on the characters in the token. The combination of those features represents a token in our feature space before being used to train the model. The features used in this experiment are based on the simple features mentioned in (Savelka, 2017). Some sample features are IsLower, IsUpper, IsSpace (see Appendix A 2 for feature sample). Using sklearn_crfsuite CRF, the model was trained using a gradient descent L-BFGS method with a maximum of 100 iterations with L1 (0.1) and L2 (0.1) regularization.

7.2 CRF Results

Predicted	Actual		
	B	I	L
B	6738	964	72
I	469	193810	481
L	95	1072	7128

Table 5 - CRF Results Confusion Matrix

	Precision	Recall	F1-Score	Support
B	0.923	0.867	0.894	7774
I	0.990	0.995	0.992	194760
L	0.928	0.859	0.892	8295
Weighted Avg	0.985	0.985	0.985	210829
Weighted Avg Excluding 'I'	0.925	0.863	0.893	16069

Table 6 - CRF Results Report

The CRF model gave us a weighted average F1-Score of 0.985 (Table 6) including the predictions for tokens "I" (Inside) the sentences. Excluding this, we get 0.893. The precision on "L" (Last) labels for the CRF model is acceptable at 92.8%.

As indicated in the frequency distribution (Section 4), the "I" (Inside) labels are the majority of the tokens. The huge number of "I" (Inside) presents an imbalance for this classification task but excluding "I" (Inside) for learning, we would lose the tendencies of the corpus. Keeping the "I" (Inside) during training would preserve the semantics of the sentences.

7.3 CRF SBD Errors

Here are some of the CRF's most common errors:

1. Citations as sentences

Sentence segmentation as labeled:

```
Franklin also moved to dismiss eleven of the fourteen copyright infringement counts on the ground that Apple failed to comply with the procedural requirements for suit under
17 U. S. C. § § 410, 411.
< 714 F. 2 d 1245 >
```

CRF's segmentation:

```
Franklin also moved to dismiss eleven of the fourteen copyright infringement counts on the ground that Apple failed to comply with the procedural requirements for suit under 17 U. S. C. § § 410, 411.
< 714 F. 2 d 1245 >
```

2. Semi-colon or colon as a sentence ending.

Sentence segmentation as labeled:

```
Defendants Simon Blitz and Daniel Gazal are the sole shareholder
s of defendants Cel - Net Communications, Inc. (" Cel - Net ");
The Cellular Network Communications, Inc., doing business as CNCG ("CNCG"); and SD Telecommunications, Inc. ("SD Telecom").
```

CRF's segmentation:

```
Defendants Simon Blitz and Daniel Gazal are the sole shareholders of defendants Cel - Net Communications, Inc. (" Cel - Net"); The Cellular Network Communications, Inc., doing business as CNCG ("CNCG"); and SD Telecommunications, Inc. (" SD Telecom ").
```

8 Neural Networks Experiment

After CRF, token context gives a significant performance gain for detecting sentence boundaries. The imbalance of the class labels is an inherent characteristic of the SBD task because sentence endings would occur at a rate, we see in the distribution frequency in written legal text for however many labeled training examples.

We experimented with a deep learning neural network representing sentence tokens as a fixed dimensional vector that encoded the context of the text using word embeddings. Gensim word2vec (Mikolov 2013) was trained on all the Adjudicatory decision data pre-processed as tokens as described in Section 4 in 200 epochs. We used an embedding size of 300 using the skip-gram model with negative sampling. Please see Appendix A 3 for the Gensim Word2vec parameters that were used.

8.1 Neural Network Training

The neural network was trained using the training data, pre-processed as described in Section 4. Each sentence token is represented as the concatenation of the vectors of word2vec embedding using a 3-word window plus 8 features (See Appendix A 4) similar to the one we used in CRF experiment, which will be the input vectors to the network.

The neural network model's architecture is a stack of Bi-Directional LSTM with a softmax output layer.

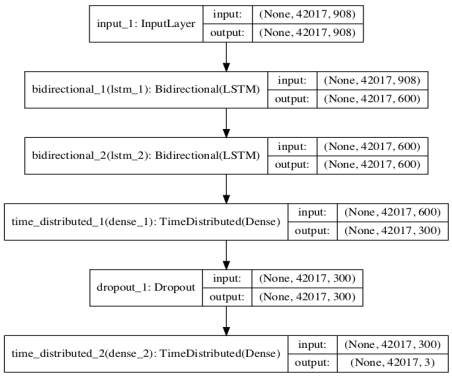

Figure 1- Neural Network Architecture

The model was trained for 40 epochs using an Adam optimizer with learning rate set to 0.001 and categorical cross-entropy as the loss function, a validation split of 15% and shuffling. We opted to use categorical cross-entropy because of its tendency to perform well on an imbalanced training set. Early stopping was also employed for up to 38 out of the 40 epochs (model training will stop if we see the model's loss increase after 38 epochs). The model was tested using the test data set.

8.2 Neural Network Results

Predicted	Actual		
	B	I	L
B	6993	668	113
I	816	193158	786
L	95	870	7330

Table 7 - Neural Network Results Confusion Matrix

	Precision	Recall	F1-Score	Support
B	0.885	0.900	0.892	7774
I	0.992	0.992	0.992	194760
L	0.891	0.884	0.887	8295
Weighted Avg	0.984	0.984	0.984	210829
Weighted Avg Excluding 'I'	0.888	0.891	0.890	16069

Table 8 - Neural Network Results Report

8.3 Neural Network SBD Errors

1. Periods took on a different context.

   Sentence segmentation as labeled:

   TRW Inc. v. Andrews, 534 U. S. 19 (2001)

   Supreme Court of the United States
   Filed: November 13th, 2001
   Precedential Status: Precedential

   NN's Segmentation:

   TRW Inc. v.
   Andrews, 534 U. S. 19 (2001) Supreme Court of the United States Filed: November 13th, 2001 Precedential

   Sentence segmentation as labeled:

   See 15 U. S. C. § § 1681 n, 1681 o (1994 ed.). [2]
   The facts of this case are for the most part undisputed.

   NN's Segmentation:

   .) . [2] The facts of this case are for the most part undisputed.

2. Specific tokens as sentence endings, e.g., "the," "With."

   Sentence segmentation as labeled:

   With him on the briefs was Harold R. Fatzer, Attorney General.
   John W. Davis argued the cause for appellees in No. 2 on the original argument and for appellees in Nos. 2 and 4 on the reargument.
   H. Albert Young, Attorney General of Delaware, argued the cause for petitioners in No. 10 on the original argument and on the reargument.
   With him on the briefs was Louis J. Finger, Special Deputy Attorney General.

   NN's segmentation:

   With
   him on the briefs was Harold R. Fatzer, Attorney General. John W. Davis argued the cause for appellees in No. 2 on the original argument and for appellees in Nos. 2 and 4 on the reargument. H. Albert Young, Attorne

```
y General of Delaware, argu
ed the
cause for petitioners in N
o. 10 on the original argum
ent and on the reargument.
With him on the briefs was
Louis J. Finger, Special De
puty
```

9 Conclusion

Out of the box sentence tokenization from popular NLP libraries like NLTK maybe be good enough for most general domain NLP tasks. However, in the legal domain, a Punkt model needs to be trained and updated to have reasonable performance on SBD especially for use with large bodies of legal text. The custom abbreviations with the Punkt model that we use are a product of the legal expertise within our organization. Without such expertise, labeled legal domain text can be used to train several algorithms to do SBD on legal text.

The CRF approach proves to be the most practical approach after comparing the results of our experiment. The neural network model's performance on token classification did not translate to a better SBD compared to the CRF Model. Ease of training and testing are an advantage of using the CRF approach. There is room for future improvement for both the CRF and the neural network approaches. For CRF, collocation features might be helpful. A different word embedding like BERT or weight balancing might improve the performance for the deep learning neural network model.

The publicly available data used in the experiments above are limited and constrained, but it is a good starting point. There is a lot more variety of legal text that was not represented in the data — for example, tax documents, dockets, and headnotes.

References

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

Walker, D. J., Clements, D. E., Darwin, M., & Amtrup, J. W. (2001, September). *Sentence boundary detection: A comparison of paradigms for improving MT quality.* In Proceedings of the MT Summit VIII (Vol. 58).

Riley MD. *Some applications of tree-based modeling to speech and language*. Proc Work Speech Nat Lang. 1989;(2):339–52.

Gillick, D. (2009). *Sentence boundary detection and the problem with the US. In Proceedings of Human Language Technologies*: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers (pp. 241-244).

Kiss, T., & Strunk, J. (2006). *Unsupervised multilingual sentence boundary detection*. Computational Linguistics, 32(4), 485-525.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Griffis, D., Shivade, C., Fosler-Lussier, E., & Lai, A. M. (2016). *A quantitative and qualitative evaluation of sentence boundary detection for the clinical domain.* AMIA Summits on Translational Science Proceedings, 2016, 88.

Liu, Y., Stolcke, A., Shriberg, E., & Harper, M. (2005). *Using conditional random fields for sentence boundary detection in speech.* In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05) (pp. 451-458).

Savelka, J., Walker, V. R., Grabmair, M., & Ashley, K. D. (2017). *Sentence boundary detection in adjudicatory decisions in the united states.* Traitement Automatique des langues, 58(2), 21-45.

Palmer, D. D., & Hearst, M. A. (1997). *Adaptive multilingual sentence boundary disambiguation.* Computational Linguistics, 23(2), 241-267.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). *Distributed representations of words and phrases and their compositionality.* Advances in neural information processing systems, 3111-3119.

A Appendices

1. Custom Abbreviation List

```
['sec','jan','feb','mar','apr','m
ay','jun','jul','aug','sep','sept
','oct','nov','dec','b.f','k.o','
l.b','h.e','h.r','o.j','n.j','u.n
','s.b','p.a','s.f','h.b','e.o','
w.s',
'g.i','p.s','g.m','p.c','m.e','a.
w','m.d','m.a','b.s','j.d','b.a',
'i.q','e.r']
```

2. CRF Sample Feature

```
['bias','0:lower=,','0:sig=,','0:len
gth=1','0:islower=false','0:isupper=
false','0:istitle=false','0:isdigit=
false','0:isspace=false','-3:BOS','-
2:lower=patrick','-2:sig=CCCCCCC','-
2:length=long','-2:islower=false','-
2:isupper=true','-
2:istitle=false','-
2:isdigit=false','-
2:isspace=false','-
1:lower=maloney','-1:sig=CCCCCCC','-
1:length=long','-1:islower=false','-
1:isupper=true','-
1:istitle=false','-
1:isdigit=false','-
1:isspace=false','1:lower=on','1:sig
=cc','1:length=2','1:islower=true','
1:isupper=false','1:istitle=false','
1:isdigit=false','1:isspace=false','
2:lower=behalf','2:sig=cccccc','2:le
ngth=normal','2:islower=true','2:isu
pper=false','2:istitle=false','2:isd
igit=false','2:isspace=false','3:low
er=of','3:sig=cc','3:length=2','3:is
lower=true','3:isupper=false','3:ist
itle=false','3:isdigit=false','3:iss
pace=false']
```

Legend:

-3 = before
-2 = before
-2 = before
0 = current token
1 = after
2 = after
3 = after
bias = string constant
BOS = Beginning of Sentence
EOS = End of Sentence
lower= token in lowercase
sig= word shape of token
    C=upper case character
    c=lower case character
    D=digit
length= length of token
    (< 4)= str(len(token))
    (>=4 token <=6) = "normal"
    (>6) = "long"
islower = binary feature which is set to true if all token characters are in lower case
isupper = binary feature which is set to true if all token characters are in upper case
istitle = binary feature which is set to true if the first token character is upper case the rest is lower case
isdigit = binary feature which is set to true if all character tokens are digits
isspace = binary feature which is set to true if all token characters are whitespace

3. Gensim Word2Vec training parameters

```
sg=1, hs=1, window=5, min_count=100,
workers=4, negative=10, ns_exponent=1
```

sg = skip-gram model
hs = use hierarchical softmax
min_count = ignores all words with total frequency less than this
workers = no. of worker threads
negative = negative sampling
ns_exponent = negative sampling distribution value of 1.0 samples are proportional to the frequencies

4. Neural Network input additional Features

isUpper = binary feature which is set to true if all token characters are in upper case
isLower= binary feature which is set to true if all token characters are in lower case
isDigit= binary feature which is set to true if all character tokens are digits
isSpace= binary feature which is set to true if all token characters are whitespace
isPunctuation= binary feature which is set to true if the token is a punctuation
Next Word Capitalized = binary feature which is set to true if the next word's first character is capitalized
Previous Word Lower= binary feature which is set to true if the previous word's first character is in lower case
Previous Word Single Char= binary feature which is set to true if the previous word a single character