LDSI W2021 Project Survey Report

Name: Isaac Misael Olguín Nolasco
Gloss ID: tum_ldsi_22

Summary

The new search engine for the BVA decisions system required by the US-BVA aims to process decisions and extract the reasons why they were considered for a granted, denied, or remanded judgment. To do so, the use of Natural Language Processing has been considered and it is informed all details of pre-processing, word embedding tool functionality, linear and non-linear classifiers as well as different sentence segmenters that were applied. It is drawn the big picture of the whole process that must be considered to create the desired system and explain benefits, potential drawbacks, and challenges that the development team would eventually face. Moreover, this document is presented as a tool of decision-making.

Introduction

The Consultancy has been hired by the US Board of Veteran's Appeals (BVA) in order to develop the so-called: New Search Engine for BVA decisions. In conformity with the development and quality policies, it is mandatory to conduct and document the feasibility, technological and risks analysis which provides all the involved staff including the development team, with enough insights about the overview, requirements, and goals of the project. This document, as well as the report of the survey literature conducted by the data scientists, show the results of the pilot experiment, and inform about the possible opportunity areas in which resources should be invested.

Information about the cases of the BVA is documented and stored in files that do not contain a specific template. Having been checked examples of those files, it can be seen a kind of traditional structure, which is not mandatory, and depends on the judge, region, or state whether they add data or do not consider fields. Each case contains sentences that can be classified into 14 different classes (this classification was conceived from the work in [1]).

1) Case header	8) Procedure
2) Case footer	9) Evidence
3) Case Issue	10) Evidence based or intermediate finding
4) Conclusion of law	11) Evidence based reasoning
5) Legal rule	12) Legislation and policy
6) Header	13) Policy based reasoning
7) Citation	14) Remand instruction

Table 1: Types/classes of sentences in cases of the BVA

To conduct the pilot, two datasets of 30K and ~101K documents were received and a small sample of 141 documents was explored to conduct an initial annotation study. This produced annotated sentences classified into the types listed in Table 1. From the analysed files, 70 cases have been granted whereas 71 have been denied; there are no remanded cases in this data[1]. The small dataset was split into 3 sets (training, development, and test). Both dev and test contain 14 decision cases (7 granted and 7 denied). Information of specific document IDs for dev and test sets can be found in Table 8.

As it has been stated, the goal of the project is to split each document content into sentences which can be classified into classes in Table 1. Therefore, the following section explains the approaches that were taken into account to decide whether a sentence segmenter has to be developed or the possibility to use one that is already in the market.

Sentence Segmenter

Two main sentence segmenters are already in the market and can be applied in the system: SpaCy's implementation and a law-specific segmenter proposed by Savelka et al. in [2]. Both

[1] Granted or denied cases might also contain "remand instructions".

implementations report promising metrics and are already available to be used. Therefore, this analysis describes the functionality and results of the implementation on the dataset.

SpaCy's standard English sentence segmenter

SpaCy is a free open-source library for NLP in Python with a standard English tokenizer, tagger, parser, and entity recognizer. It is capable to segment a whole text into sentences. In order to determine whether it may be applied to the US-BVA cases, the 113 annotated documents (corresponding to the training set) were considered as ground truth and compared against the sentences provided by SpaCy's segmenter.

Total docs	Total # sentences	Accuracy	Precision	Recall	F1-Score
113	15,349	45.14%	59.9%	59.37%	58.99%

Table 2. Metrics on SpaCy segmenter (training set)

There were chosen three documents whose metrics are low (the worst) and affect the average of the overall result.

Document	True Pos.	False Pos.	False Neg.	Accuracy	Precision	Recall	F1-Score
0811472.txt	54	114	82	21.6%	32.14%	39.70%	35.53%
1435140.txt	8	12	17	21.62%	40%	32%	35.55%
1221850.txt	21	23	30	28.38%	47.73%	41.18%	44.21%

Table 3. Metrics on the SpaCy segmenter (specific documents of the training set)

From the previous analysis, the following behavior of the segmenter was concluded:

- It is used to split sentences when it finds a sequence of spaces, carriage return characters, and newline characters. However, it does not remove such characters which are not important and are not to be considered for the classification task. i.e., they are not part of a specific format and are irrelevant for the task. Sometimes the length of such characters exceeds the threshold specified by the current analysis implementation (+- 3).
- Headers such as "`THE ISSUE`", "`INTRODUCTION`", "`ATTORNEY OF THE BOARDS`", "`REPRESENTATION`", etc. are included in other sentences and they should be split as single sentences.
- Sequences such as "I.", "II."; "1.", "2.", "3." are considered sentences. They should not be split of the following data since they are listing bullet points.
- Even though there are specific headers, cases do not follow a specific template and it depends on how the judge wants to write the report is the way it will be structured. In the specific case of the file "`0811472.txt`" there are more headers ("`Analysis`"). Nevertheless, having found these keywords, they do not represent a pattern unless there are newlines before and/or after.
- It was reported as a False Positive in the following case (`1435140.txt`).

False Positive	The RO is found to have complied with the Board's rem and instructions
Original sentence	The case was remanded for further development in April 2011, June 2012, June 2013, and December 2013. The RO is found to have complied with the Board's remand inst ructions.

  This should not be considered as incorrect, but the sentence was not appropriately split during the manual analysis.

Given the previous results, SpaCy offers the possibility to make some changes or add extensions/exceptions to the sentence splitter. A language component(i) was added to the pipe with the aim of splitting sentences when a header is found. Additionally, it was added a change whose goal is to avoid splitting a sentence when a numbered list is observed. On the one hand, accuracy and precision seem to be equal. On the other hand, recall shows an

improvement of 6-7 points, which should be interpreted as a reduction in the number of true split sentences that had not been matched. However, the biggest enhancement occurs when there are spaces before and after each sentence and those characters are removed. This does not change the content but drops all unnecessary characters in the beginning and in the end of it. Metrics are reported in Table 5.

Improvement	Accuracy	Precision	Recall	F1-Score
(i) Headers	45.98%	59.99%	66%	62.70%
(ii) Control on numbered lists	45.97%	59.93%	66.05	62.76%
(iii) Removing spaces in the beginning and end on sentences produced by Spacy	79.38%	86.80%	90.38%	88.42%

Table 4. Comparison of metrics when extension/exceptions are added

Law-specific sentence segmenter

Savelka et.al. [2] produced a law-specific sentence segmenter based on decisions in the United States, i.e., their implementation is capable to extract information from decisions in the form of sentences. The experiment was tested against the same training data giving the following results.

Total docs	Total # sentences	Accuracy	Precision	Recall	F1-Score
113	15,349	81.76%	83.67%	97.13%	89.57%

Table 5. Metrics on Savelka's sentence segmenter (training set)

It was done a similar analysis on the results and from it, there were chosen 3 specific cases whose metrics are the worst from the dataset.

Document	True Pos.	False Pos.	False Neg.	Accuracy	Precision	Recall	F1-Score
1014839.txt	95	166	2	36.13%	36.40%	97.94%	53.07%
0842824.txt	117	78	12	56.52%	60%	90.70%	72.22%
1107494.txt	103	47	9	64.78%	68.67%	91.96%	78.63%

Table 6. Metrics on the Savelka's segmenter (specific documents of the training set)

From the analysis above, the following points should be notice:

- The documents "1014839.txt" and "0842824.txt" are cases of a service connection bilateral hearing loss disabilities, whose texts contain a list of frequencies on hearing disability thresholds and the corresponding levels of the veterans' disabilities. According to how these data was managed by the team, it was linked to "Evidence" or "Evidence based or intermediate finding". It means the sentence segmenter considers each number as a new sentence while the ground truth considers it as a sentence built on several frequency numbers.
- The third document "1107494.txt" is a special case that contains several new lines in the original text. It is believed that such lines are interpreted by the segmenter as new lines and hence, it splits the original text into more sentences than the number reported by the analysis (considered as ground truth). In the following Table 9, it is exhibited one paragraph and its corresponding sentences, and the ones produced by Savelka's segmenter. It is assumed that during the creation and storage of the file, additional newlines were added, probably to give a specific format to it.

Original text (just one paragraph)
An April 2010 rating decision granted service connection for acid reflux and assigned a noncompensable evaluation effective November 4, 2004. As the issue of service connection for a gastrointestinal condition has been granted, it is no longer part of the Veteran's appeal. See Grantham v. Brown, 114 F.3d 1156 (Fed. Cir. 1997) (where an appealed claim for service connection is granted during the pendency of the appeal, a second Notice of Disagreement must thereafter be timely filed to initiate

appellate review of "downstream" issues such as the compensation level assigned for the disability or the effective date of service connection)	
Sentences marked as ground truth	**Sentences produced by the segmenter**
An April 2010 rating decision granted service connection for acid reflux and assigned a noncompensable evaluation effective November 4, 2004.	An April 2010 rating decision granted service connection for acid reflux and assigned a noncompensable evaluation effective
	November 4, 2004.
As the issue of service connection for a gastrointestinal condition has been granted, it is no longer part of the Veteran's appeal.	As the issue of service connection for a gastrointestinal condition has been granted, it is no longer part of the Veteran's appeal.
	See Grantham v. Brown, 114 F.3d 1156
See Grantham v. Brown, 114 F.3d 1156 (Fed. Cir. 1997) (where an appealed claim for service connection is granted during the pendency of the appeal, a second Notice of Disagreement must thereafter be timely filed to initiate appellate review of "downstream" issues such as the compensation level assigned for the disability or the effective date of service connection).	(Fed. Cir. 1997) (where an appealed claim for service connection is granted during the pendency of the appeal, a second Notice of Disagreement must thereafter be timely filed to initiate appellate review of "downstream" issues such as the compensation level assigned for the disability or the effective date of service connection).

Table 7. Example of Savelka's segmenter on a sample's paragraph with bad metrics

In addition, this file also contains numbered lists which are split then, more False Positives are reported. Some examples of this are "`1. All known and…`", "`2. The Veteran's…`", and "`3. A December…`". A deeper examination was conducted, the conclusion is that such sentences have been divided due to the existence of two spaces between the point character and the first word which it is assumed as a new sentence. Moreover, it is believed that this document is an outcome of a copy-paste from another source and is required an inspection on the Savelka's segmenter to know whether an improvement on it can be done.

Considering both segmenters (SpaCy and Savelka's implementations), it is concluded the use of the Savelka's segmenter given the good results on recall of 97.13% (small number of true splits that are not matched) and precision of 83.67% (small number of generated splits that are not matched).

Pre-processing

One of the most important steps in the development of a project that implements ML, DL, or NLP is the preparation of data since it is the input of tasks like analysis, predictions, classifications, etc. To prepare the information for the project and once Savelka's sentence segmenter was chosen, the whole unlabelled corpus[2] was processed to know the number of sentences that the corpus contains. In total, there are 3'360,495 sentences reported from the 30K cases.

Taking into consideration the way Noguti et al. [3] leads the pre-processing step, it was followed the same idea considering tokenization, lowercasing, punctuation removal, numeral normalization, lemmatization, and non-ASCII character removal, except name normalization

[2] The unlabeled corpus contains 30K cases (10K are granted, 10K are denied and 10K are remanded).

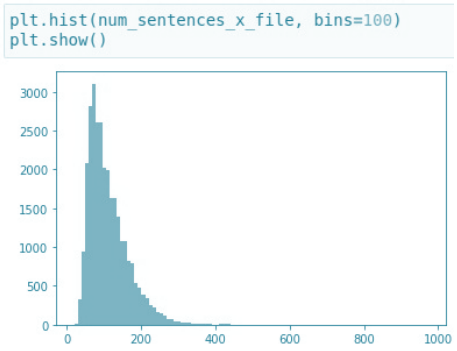

Figure 1. Number of sentences per file

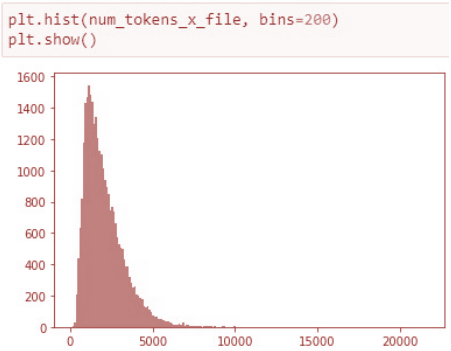

Figure 2. Number of tokens per file

Both graphs show consistency. The bigger number of sentences, the bigger the number of tokens.

and Part of Speech tagging. (Removing stop words was also conducted but results without removing them seems to be more promising. It is assumed that this is due to the lack of more annotated data and then it is considered as part of further work).
The first approach considered the implementation of a custom tokenizer, but after a while, there were faced many challenges to achieve lemmatization and other benefits that already offers SpaCy. Therefore, the so-called processing pipeline is considered for pre-processing.

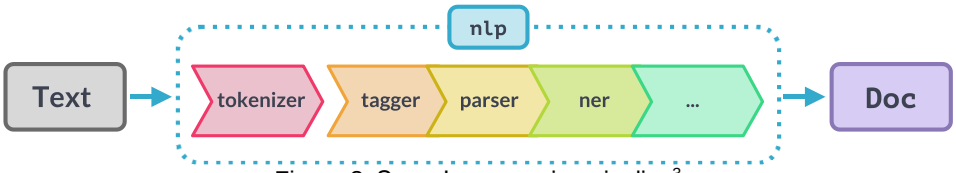

Figure 3. Spacy's processing pipeline[3]

The pipeline contains at least the following components:

- Tokenizer: which segments text into tokens. It creates the Doc object that is returned once the pipeline has been executed by all components.
- Tagger: it sets part-of-speech (PoS) tags.
- Parser: this assigns dependency labels.
- Entity recognizer (ner): which detects and labels entities
- Lemmatizer: it assigns lemmas or base forms.
- Text categorizer: it sets document labels, and
- Custom components: performs additional attributes or methods defined explicitly. This has been already used for the sentence segmenter.

First, the text of each case is transformed into *lower case* characters. This is performed for simplicity and to maintain the consistent flow during the task classification. When different entities are found, for example, a capitalized version and another that is completely lower-case, this could deal with sparsity issues, moreover when the dataset is not large, as it is in this pilot. Then applying this transformation allows reducing the sparsity. *Tokenization* splits sentences into several tokens which are categorized by SpaCy into classes of words[4]: adjective, adposition, adverb, auxiliary, coordinating conjunction, determiner, interjection, noun, numeral, particle, pronoun, proper noun, punctuation, subordinating conjunction, symbol, verb, and other; this allows the pre-processing to easily remove all punctuation characters. Although the number of characters that are being used by people to transmit information such as emojis, short word versions or special characters has increased, it is not the case of the business model. Texts to be analysed by the engine would contain law-specific language and punctuation could add unnecessary noise. This is applied also to the non-ASCII characters whose presence introduces noise to the model, then they are removed. *Numbers normalization* should be performed; there is a huge diversity when writing numbers like years, dates, etc., all they represent quantity, and hence, they can be changed to give the appropriate

[3] https://spacy.io/usage/processing-pipelines
[4] https://spacy.io/api/token & https://universaldependencies.org/u/pos/

information to the model for the learning process. It is followed the strategy proposed by Grabmair[5] which changes any number to <NUM#> where # represents the length of the number. For instance, if there is 2022, it is transformed to <NUM4>. Normalization should not be considered as an unimportant task, since according to Satapathy et. al. [4] it could improve the accuracy of metrics. *Lemmatization* allows the reduction of words to their base form. It aims to remove inflections. Even though Camacho-Collados et. al. [5] claims lemmatization has no significant impact on the accuracy of text classification, it was decided to make use of the lemmas such that the complex law language could be reduced to a smaller vocabulary and easier for the model to learn. *Stop words* are a set of commonly used words in a language, in the specific case of the English language, instances of them are “a”, “the”, “is”, “are”, etc. The idea behind removing those words is to extract information that is not relevant words to focus on the most important ones that remain in a text. There were conducted two test cases: texts with stop words and texts without stop words which are not discussed in this brief report because the one with stop words shows better results.

Comparison of files with max and min sentences/tokens			
Case	Lines	Sentences	Tokens
0734716	2,717	974	21,655
0727943	60	20	125

Due to the huge difference between the documents reported in the table on the left, both files were scanned and although the first document contains lots of data and is consistent with the number of sentences obtained by the segmenter, the second lacks information and this allows us to conclude that either the case is valid with these few details or the document is incomplete and therefore, this could have an impact on the model and then represent a bias for the learning process of the new engine.

The following list reports the IDs of the documents that were chosen randomly from the 141 cases. As it was stated in the Introduction part, 14 cases form the dev set and 14 the test set (7 are granted and 7 are denied respectively).

Dev	Granted	'61aea55f97ad59b4cfc41308', '61aea55d97ad59b4cfc412be', '61aea55c97ad59b4cfc412ac', '61aea55e97ad59b4cfc412fd', '61aea55f97ad59b4cfc41335', '61aea55c97ad59b4cfc412ae', '61aea55f97ad59b4cfc41336'
	Denied	'61aea57497ad59b4cfc413d1', '61aea57097ad59b4cfc41355', '61aea57397ad59b4cfc41393', '61aea57497ad59b4cfc413d7', '61aea57097ad59b4cfc41360', '61aea57097ad59b4cfc41368', '61aea56f97ad59b4cfc41343'
Test	Granted	'61aea55e97ad59b4cfc412de', '61aea55e97ad59b4cfc412eb', '61aea55f97ad59b4cfc41328', '61aea55f97ad59b4cfc4130e', '61aea55f97ad59b4cfc41323', '61aea55f97ad59b4cfc41337', '61aea55f97ad59b4cfc41322'
	Denied	'61aea57497ad59b4cfc413b3', '61aea57497ad59b4cfc413da', '61aea57497ad59b4cfc413d8', '61aea57497ad59b4cfc413d2', '61aea57497ad59b4cfc413ba', '61aea57097ad59b4cfc41358', '61aea57397ad59b4cfc413a5'

Word Embeddings

Word embeddings are capable to capture the context of a word in a document. Concisely, embeddings are vector representations of a particular word within a document. The goal behind this technique is that given words with similar context, they might be in close spatial positions. Mathematically speaking, it is required the compute the cosine similarity because proportional vectors have a cosine similarity close to 1. FastText[6] is an open-source, free, lightweight library for efficient text classification and representation learning; it was used during the implementation to produce the embeddings from the unlabelled corpus’ tokens. The

[5] M. Grabmair, Classification workshop LDSI WS 2021 TUM
[6] https://fasttext.cc/

embedding was created[7] with a 100-dimensional vector model, minimum word occurrent count of 20, 10 epochs, and 4 threads).

The number of words read by FastText, when stop words are not removed, is 63M words and the vocabulary size is 12,625. A simple way to inspect the quality of the vectors that are produced by FastText is looking by the nearest neighbours because it gives an intuition of the semantic information of a specific vector. This tool offers the possibility to fetch the 10 nearest neighbours which were used to perform the retrieval of the following words: “veteran”, “v.”, “argues”, “ptsd”, “granted”, “korea”, “holding”, “also”, “diagnosed”, “injured”, “surgery”, “accident”, “disorder”, “posttraumatic”, “depression”, “anxiety”.

In the case of the word “ptsd”, whose meaning is associated to the “post-traumatic stress disorder”, their 10-nearest neighbours are depressive, pstd, anxiety, mdd, depression, dysthymia, dysthymic, psychiatric, bipolar, and agoraphobia. A simple study case is the file “1329319.txt”, in which the sentence “depressive disorder” occurs 21 times and it contains just a couple instances of the word “ptsd” (its appearance occurs on URLs). On the other hand, bipolar, agoraphobia, dysthymia, and major depressive disorder (MDD) are other types of disorders, whereas anxiety and depression are symptoms of all these disorders. Finally, “pstd” and “dysthymic" are not correctly written but have been interpreted as other entities and their appearance happens after lots of occurrences on the unlabeled data with the same context.

When the word embedding creation was executed for the pilot, it was instructed to do it using skip-gram, which is a word2vec method to construct an embedding and which takes a word as input and tries to predict its context. The nearest neighbor of the word “veteran” is the appellant. This makes complete sense since it’s the veteran the appellant of the case. Other neighbors are predictable because are possessive syntactic structures, articles, nouns. However, the word “march” appears as the 10-nearest neighbor and it is not possible to infer its appearance.

Most of the neighbors for the word “granted” are synonyms, antonyms, or derivations of the lemma grant, such as “grant” or “granting”. Another important neighbor is “tdiu”, which is the abbreviation of Total Disability Individual Unemployability and represents a rating for veterans who cannot work because to their service-connected disability and whose presence is due to the number of instances that have been granted given this TDIU.

The power of embeddings can also be seen with the results of:

- injure. From which obtained words such as reinjure, twist, hurt, hit, sustain, and even throw or jump are expected, but basketball, volleyball, and jump are also among the neighbors, which allows to know the high frequency of disabilities are result of playing those sports.
- surgery, whose neighbors could show a list of associated words with surgical interventions, such as capsulotomy, removal, grafting, resection, exploration, complication, and arthroscopic. Nonetheless, one word that is under investigation is “underwent”, since it is the past tense of the verb undergo and should not be fetched as a neighbor given the implementation of the tokenizer built on SpaCy’s pipeline and which is capable to retrieve lemmas of words.

Expected neighbors of the word “korea” are korean, dmz, demilitarized, and demilitarize which represent the nationality and the unique area in the world between North and South Corea called “demilitarized zone”. More interesting is the presence of Germany, Panama, and Vietnam. This might not be of specific associations among all countries but word analogies are built by FastText within the word embeddings.

This leads to the conclusion that the embedding has successfully created the desired similarity of words and the capability of such embeddings to not just link context but also infer analogies.

Training & Optimizing Classifiers

To produce the best model, there are two possible options to transform words into vectors, one was the approach covered in the previous section when it was discussed embeddings, another option is the technique “Term frequency-inverse document frequency

[7] ./fasttext skipgram -input [inputFileWithTokens].txt -output [pathOutput] -verbose 2 -minCount 20 -dim 100 -epoch 10 -thread 4

(TFIDF)". This alternative is a numerical statistic approach that aims to exhibit how important a word is to a document. The TFIDF [8]value increases proportionally to the number of times a word appears and it is offset by the number of documents in the corpus.

Top 10 TFIDF values of tokens in a sentence[9]		
Feature	**TFIDF**	**Sentence**
undiagnosed illness for due deny carolina columbia excess gastrointestinal south posttraumatic	0.303441 0.284814 0.224093 0.217028 0.201969 0.188310 0.184941 0.182022 0.177145 0.166871 0.165546	In October 2008, the Board of Veterans' Appeals (Board) denied service connection for posttraumatic stress disorder (PTSD) and vision problems due to an undiagnosed illness; denied a rating in excess of 20 percent for low back strain, and remanded the issues of service connection for a gastrointestinal condition and headaches due to an undiagnosed illness to the Department of Veterans Affairs (VA) Regional Office in Columbia, South Carolina (RO) for additional development.

Both transformation of words into vectors (TFIDF and embeddings) were applied and both linear and non-linear classifiers were trained on the train set and tested against the dev set. It was sought to make the models learn how to generalize the information so that they learn how to classify different sentences with a very small dataset. The classifiers that were considered are Linear support vector machine, and logistic regression on the side of linear models, while decision trees, random forest, and support vector machines with either polynomial or radial kernel were on the side of non-linear classifiers. The first method that was used to train the classifiers, was a manual approach in order to gradually tun the hyperparameters. It was started with the default values of each classifier. However, finding the best combination following this way is not guaranteed and it is time-consuming, unstructured, inefficient, and tedious. Fortunately, scikit-learn[10] contains techniques for the hyperparameter tuning process: grid and random search, from which the second one was chosen given its flexibility to define a range or a list of possible values that must be considered. This procedure does not only allow to choose them randomly, but it also does not require human intervention, i.e., it is a task that can be delegated to the computer to find the best combination. The parameters used to analyse classifiers' behaviour and metrics are max_iter, dual, loss, penalty, and C (regularization parameter) for the linear models, and max_depth, min_samples_split, n_estimators, kernel, degree, gamma, C, splitter, criterion on the side of non-linear models according to its definition on the scikit-learn website.

Each classifier was trained on different hyperparameters[11]. The best classifiers and their metrics are shown in Table 8. To decide the best option among the different classifiers, each model was trained, and their corresponding report metrics were analysed to know which one gives the best results, i.e., precision, recall, and F1-score should be close to one. Even though some classifiers such as decision trees or random forest achieved the highest score for the metrics in the training set, they had poor results when testing against the dev set. This must be interpreted as they were memorizing but were not capable of generalizing well. Once the best models were chosen, test-set was used for proving their behaviour which was consistent with dev-set metrics.

Error Analysis

Confusion matrices of the best TFIDF and embedding-based models on the dev set can be seen in Figures 4 and 5 respectively. The first problem that needs to be addressed is that there are at least 3 remanded instructions in the training set while there are instances of such class neither in the dev nor the test set. This may conduct to an arduous task for the engine to classify correctly. Second, a similar challenge can be seen for the "policy-based reasoning" class; there are 4 and 1 cases in the dev and test sets, and any of them are classified correctly.

[8] https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089
[9] Examples of sentences on train, dev and test set can be found in the file located at resources/TopTidfdTokensInSentence.pdf
[10] https://scikit-learn.org/
[11] The list of hyperparameters can be found in resources/HyperparametersClassifiers.pdf

TFIDF	Embeddings
SVC: {kernel="rbf", gamma="scale", C=1, max_iter=-1}	SVC: {kernel="rfb", gamma="scale", C=4}

TRAIN:

	precision	recall	f1-score	support
CaseFooter	1.00	0.99	1.00	113
CaseHeader	0.99	0.97	0.98	115
CaseIssue	0.97	0.99	0.98	114
Citation	0.99	1.00	0.99	1889
ConclusionOfLaw	0.95	0.93	0.94	270
Evidence	0.91	0.98	0.94	3720
EvidenceBasedOrIntermediateFinding	0.89	0.86	0.87	1171
EvidenceBasedReasoning	0.91	0.67	0.77	835
Header	0.99	0.99	0.99	1178
LegalRule	0.92	0.97	0.94	1515
LegislationAndPolicy	0.82	0.36	0.51	137
PolicyBasedReasoning	0.00	0.00	0.00	24
Procedure	0.98	0.96	0.97	1087
RemandInstructions	0.00	0.00	0.00	3
accuracy			0.94	12171
macro avg	0.81	0.76	0.78	12171
weighted avg	0.94	0.94	0.93	12171

DEV:

	precision	recall	f1-score	support
CaseFooter	0.87	1.00	0.93	13
CaseHeader	1.00	1.00	1.00	14
CaseIssue	1.00	1.00	1.00	14
Citation	0.96	0.98	0.97	250
ConclusionOfLaw	0.88	0.93	0.90	30
Evidence	0.74	0.94	0.83	381
EvidenceBasedOrIntermediateFinding	0.74	0.51	0.60	149
EvidenceBasedReasoning	0.42	0.20	0.28	93
Header	1.00	0.96	0.98	150
LegalRule	0.87	0.90	0.88	181
LegislationAndPolicy	0.71	0.31	0.43	16
PolicyBasedReasoning	0.00	0.00	0.00	4
Procedure	0.91	0.90	0.91	136
accuracy			0.84	1431
macro avg	0.78	0.74	0.75	1431
weighted avg	0.82	0.84	0.82	1431

TEST:

	precision	recall	f1-score	support
CaseFooter	1.00	1.00	1.00	14
CaseHeader	1.00	1.00	1.00	14
CaseIssue	0.93	1.00	0.97	14
Citation	0.99	0.95	0.97	291
ConclusionOfLaw	0.94	0.78	0.85	37
Evidence	0.80	0.94	0.86	540
EvidenceBasedOrIntermediateFinding	0.67	0.57	0.61	153
EvidenceBasedReasoning	0.49	0.28	0.35	123
Header	1.00	0.96	0.98	151
LegalRule	0.88	0.90	0.89	246
LegislationAndPolicy	1.00	0.14	0.24	22
PolicyBasedReasoning	0.00	0.00	0.00	1
Procedure	0.86	0.99	0.92	141
accuracy			0.85	1747
macro avg	0.81	0.73	0.74	1747
weighted avg	0.84	0.85	0.84	1747

TRAIN:

	precision	recall	f1-score	support
CaseFooter	1.00	1.00	1.00	113
CaseHeader	0.99	0.97	0.98	115
CaseIssue	0.96	0.99	0.97	114
Citation	0.99	0.99	0.99	1889
ConclusionOfLaw	0.91	0.87	0.89	270
Evidence	0.85	0.94	0.90	3720
EvidenceBasedOrIntermediateFinding	0.70	0.68	0.69	1171
EvidenceBasedReasoning	0.61	0.34	0.44	835
Header	0.99	1.00	0.99	1178
LegalRule	0.85	0.93	0.89	1515
LegislationAndPolicy	0.77	0.29	0.42	137
PolicyBasedReasoning	1.00	0.04	0.08	24
Procedure	0.94	0.94	0.94	1087
RemandInstructions	0.00	0.00	0.00	3
accuracy			0.88	12171
macro avg	0.83	0.71	0.73	12171
weighted avg	0.87	0.88	0.87	12171

DEV:

	precision	recall	f1-score	support
CaseFooter	0.93	1.00	0.96	13
CaseHeader	1.00	1.00	1.00	14
CaseIssue	0.88	1.00	0.93	14
Citation	0.96	0.97	0.96	250
ConclusionOfLaw	0.83	0.97	0.89	30
Evidence	0.76	0.92	0.83	381
EvidenceBasedOrIntermediateFinding	0.70	0.52	0.60	149
EvidenceBasedReasoning	0.62	0.28	0.39	93
Header	1.00	0.97	0.98	150
LegalRule	0.83	0.91	0.87	181
LegislationAndPolicy	0.71	0.31	0.43	16
PolicyBasedReasoning	0.00	0.00	0.00	4
Procedure	0.92	0.92	0.92	136
accuracy			0.84	1431
macro avg	0.78	0.75	0.75	1431
weighted avg	0.83	0.84	0.83	1431

TEST:

	precision	recall	f1-score	support
CaseFooter	1.00	1.00	1.00	14
CaseHeader	1.00	0.86	0.92	14
CaseIssue	0.93	1.00	0.97	14
Citation	0.98	0.97	0.97	291
ConclusionOfLaw	0.82	0.76	0.79	37
Evidence	0.82	0.92	0.86	540
EvidenceBasedOrIntermediateFinding	0.57	0.56	0.57	153
EvidenceBasedReasoning	0.47	0.25	0.33	123
Header	0.99	0.99	0.99	151
LegalRule	0.86	0.89	0.87	246
LegislationAndPolicy	1.00	0.14	0.24	22
PolicyBasedReasoning	0.00	0.00	0.00	1
Procedure	0.90	0.98	0.94	141
accuracy			0.84	1747
macro avg	0.80	0.72	0.73	1747
weighted avg	0.83	0.84	0.83	1747

Table 8. Metrics of the best classifiers on TFIDF and Embeddings

To sum up, the lack of examples of these classes is undoubtedly a source of the inaccuracy of the metrics and hence misclassification. It is assumed that the model does not have enough features that can be used to set such instances into their correct classes.

The confusion matrices also show that most misclassifications happen across the "Evidence", "Evidence-based reasoning", and "Evidence based or intermediate finding". This is a foreseeable challenge since most people find it difficult to distinguish among these classes too and lawyers or people in the legal area face these kinds of problems at the very beginning of their careers. Therefore, it should be expected the need of the legal office's help.

The following example shows the difficult task of the classifier. The first sentence fits the requirement to be considered as evidence. However, the second part adds extra information that makes the whole sentence a finding.

Case: 1030062	Prediction: Evidence	True: Evidence-based or intermediate finding
Those records reflect that on October 2003 entrance examination, the Veteran demonstrated bilateral hearing loss that qualifies as disability for VA purposes.		

On the other hand, the next examples show the other way around. The semantics of the sentences allows the reader to know that the Veteran has claimed something, but they are not statements of authoritative findings or conclusion of whether the evidence satisfies a legal rule. Hence, they should be considered as evidence.

Case: 0611114	Prediction: Evidence-based or intermediate finding	True: Evidence
Prior to this exam, the veteran denied suffering any heart problems.		
He also denied having suffered or then-suffering from heart trouble, palpitation or pounding heart, high or low blood pressure, or shortness of breath.		

During the error analysis, it was found the following scenario:

Case: 0918858	Prediction: Evidence	True: Procedure
The Veteran had active military service from March 1953 to February 1955.		

While any person may consider this sentence as evidence, it may be claimed a mistake within the classification engine. The position of the text and annotation rules must be considered. It

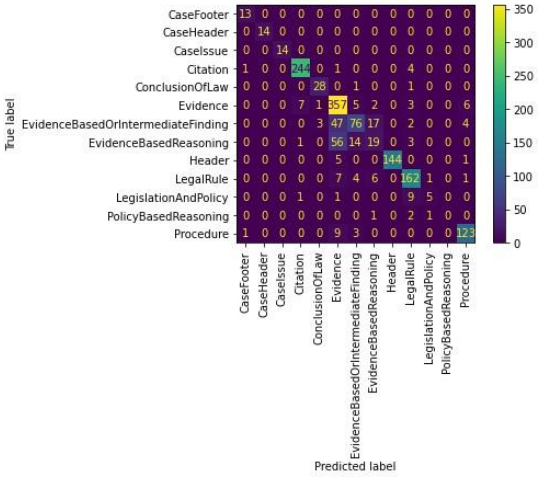

Figure 4. TFIDF – Confusion matrix on dev test

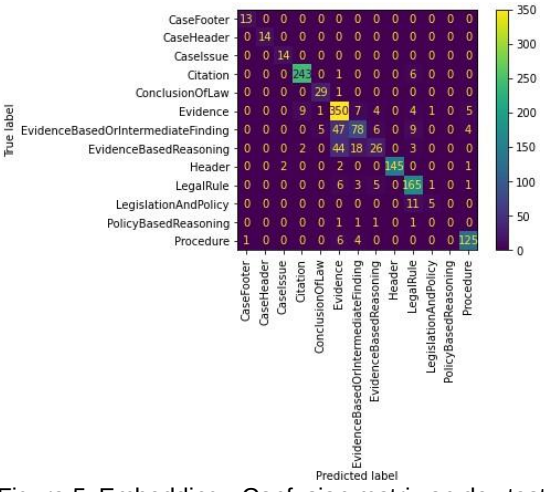

Figure 5. Embedding - Confusion matrix on dev test

is stated by the instructions of the semantic types: "*Procedure: Statements of procedural facts and formalities in case… The initial statement about the veteran's service period in the introduction should be annotated as Procedure*"[12]. It was found that many misclassifications were result of Procedure sentences. Further development should invest time on defining a way to obtain features from such parts of the text that lead to correct classification.
Different sentences from these three categories of evidence were analysed to know whether they were assigned to the correct class on the dev set and no mismatches were found.

Discussion

Once fully implemented, the pilot allows the conclusion about the importance to have a law-specific segmenter. Savelka's segmenter provides high precision and recall metrics but an investigation on numbered lists is pending, it is assumed that it can be still improved by the dev-team and lead to better results. SpaCy and scikit-learn offer numerous advantages for pre-processing, training classifiers, and finding the best combination of hyperparameters. SpaCy is a trustworthy tool whose implementation and functionality should be taken advantage of; however, the appearance of the word "underwent" when lemmatization had been already applied, left an open question about the possibility of finding other exceptions. On the side of skicit-learn, the are no objections about their functionality and the possibility to work with, but it is assumed that better metrics can be achieved if Long Short-Term Memory is implemented in the project. Word embeddings showed they are capable to build analogies and the context of words. Unfortunately, it is guessed that the length of the input was insufficient to give the classifier more features to distinguish and complete its task with better results. It is believed also that the biggest problem that the pilot and any further development face is the lack of annotated data and the need for skilled people on law-field who can produce the training dataset. It is agreed that without such trained people, it is not possible the successfully conduction of the project. Therefore, resources should be address to both IT and legal departments.

Further work should be conducted on Savelka's segmenter improvement, SpaCy's pipeline, word embeddings with a bigger vocabulary, LTSM or Bi-LTSM. Given the importance of the classification and its impact on Veteran's lives. It is reminded that the engine requires constant human supervision, which can be initially focus on the main problematic classes.

Cleaning data on a task that is natural language-based must not be underestimated since it is the main source of the system and its reason for being. Although it is a time-consuming task, it should be done with the highest priority and responsibility. Additionally, the most efficient use of memory should be considered; deactivating the Spacy's pipeline does not necessarily represent an improvement in speeding up the execution since it was tried to implement multiprocessing with Python, in addition to the multiprocessing that Spacy offers but the performance was worse.

[12] LDSI Annotation Guidelines (WS2021) v2

Pilot execution

A README file has been added to the delivery. All necessary information to run the pilot is provided there.

References

[1] Walker, Vern R., Ji Hae Han, Xiang Ni, and Kaneyasu Yoseda. "Semantic types for computational legal reasoning: propositional connectives and sentence roles in the veterans' claims dataset." In Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law, pp. 217-226. 2017

[2] Savelka, Jaromir, Vern R. Walker, Matthias Grabmair, and Kevin D. Ashley. "Sentence boundary detection in adjudicatory decisions in the United States." Traitement automatique des langues 58 (2017): 21

[3] M. Y. Noguti, E. Vellasques and L. S. Oliveira, “Legal Document Classification: An Application to Law Area Prediction of Petitions to Public Prosecution Service,” 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9207211

[4] R. Satapathy, C. Guerreiro, I. Chaturvedi and E. Cambria, “Phonetic-Based Microtext Normalization for Twitter Sentiment Analysis,” 2017 IEEE International Conference on Data Mining Workshops (ICDMW), 2017, pp. 407-413, doi: 10.1109/ICDMW.2017.59.

[5] Camacho-Collados, José and Taher Pilevar, Mohammad, “On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis”, Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pp. 40-46, 2018

```
LegalRule                          However, even
                                   assuming that there is a valid diagnosis of PTSD under the
                                   criteria of DSM-IV, service connection for PTSD also requires
                                   credible evidence of an in-service stressor.
Citation                           38 C.F.R. §
                                   3.304(f).
LegalRule                          As to the existence of a stressor, a veteran's assertions of
                                   non-combat service stressors are not sufficient to establish
                                   their occurrence.
LegalRule                          Rather, his alleged service stressors must
                                   be established by official service records or other credible
                                   supporting evidence.
Citation                           38 C.F.R. § 3.304(f); Pentecost v.
                                   Principi, 16 Vet. App. 124 (2002); Fossie v. West, 12 Vet.
Citation                           App. 1 (1998); Cohen v. Brown, 10 Vet. App. 128 (1997); Doran
                                   v. Brown, 6 Vet. App. 283 (1994).
EvidenceBasedReasoning             There is no evidence which corroborates the occurrence of the
                                   stressor alleged by the veteran.
EvidenceBasedReasoning             The Board is unable to find
                                   that the veteran's alleged service stressor has been verified
                                   by official service records or other credible supporting
                                   evidence.
EvidenceBasedReasoning             The weight of the credible evidence establishes that a
                                   stressor which might lead to PTSD did not occur in service.
EvidenceBasedOrIntermediateFinding Thus, regardless of diagnosis, service connection for PTSD
                                   may not be granted.
LegalRule                          The veteran may apply to reopen his
                                   claim in the future, by submitting independent evidence to
                                   corroborate a service stressor, or by submitting sufficiently
                                   detailed information as would permit the VA to attempt
                                   stressor verification through the service department.
Citation                           See 38
Citation                           C.F.R. § 3.159(c)(2).
EvidenceBasedOrIntermediateFinding As the preponderance of the evidence is against the claim for
                                   service connection for PTSD, the benefit-of-the-doubt rule
                                   does not apply, and the claim must be denied.
Citation                           38 U.S.C.A. §
                                   5107(b); Gilbert v. Derwinski, 1 Vet. App. 49 (1990).
Header                             ORDER
ConclusionOfLaw                    Service connection for PTSD is denied.
Citation                           K. A. BANFIELD
CaseFooter                         Veterans Law Judge, Board of Veterans' Appeals



                                    Department of Veterans Affairs
```

Example of python analyze.py 0600090.txt