Learning Sentence Embeddings in the Legal Domain with Low Resource Settings

Sahan Jayasinghe[*,1], **Lakith Rambukkanage**[*], **Ashan Silva**[*], **Nisansa de Silva**[*], **Amal Shehan Perera**[*], and **Madhavi Perera**[**]

[*]Department of Computer Science & Engineering, University of Moratuwa, Sri Lanka
[**]Parliament of Sri Lanka, Sri Lanka
[1]sahanjayasinghe.17@cse.mrt.ac.lk

Abstract

As Natural Language Processing is evolving rapidly, it is used to analyze domain specific large text corpora. Applying Natural Language Processing in a domain with uncommon vocabulary and unique semantics requires techniques specifically designed for that domain. The legal domain is such an area with unique vocabulary and semantic interpretations. In this paper we have conducted research to develop sentence embeddings, specifically for the legal domain, to address the domain needs. We have carried this research under two approaches. Due to the availability of a large corpus of raw court case documents, an Auto-Encoder model which re-constructs the input sentence is trained in a self-supervised approach. Pre-trained word embeddings on general corpora and word embeddings specifically trained on legal corpora are also incorporated within the Auto-Encoder. As the next approach we have designed a multitask model with noise discrimination and Semantic Textual Similarity tasks. It is expected that these embeddings and gained insights would help vectorize legal domain corpora, enabling further application of Machine Learning in the legal domain.

1 Introduction

Natural Language Processing (NLP) is advancing rapidly in the research domain as well as in the practical applications. Several researches have been conducted in the recent past, that have made ground breaking progress but some are yet to be discovered and applied in practical applications. It is not trivial to apply ML techniques directly on unstructured data and NLP approaches different aspects of these problems. Also the advantages of using NLP is best utilized in fields which handle large amounts of textual data. The Legal Domain is such a domain where an abundance of textual data is available, and legal corpora is growing on a daily basis.

1.1 Case Law

Case Law documents are one of the aspects which contributes to the rapidly growing textual data in the legal domain. In case law, records of past cases with their evidence arguments and judgment are kept in order to be used as reference and grounds for ongoing cases (cor, 2020). The usage of similar cases with respect to the current case as grounds, is why these documents are very important in a predictive sense. They serve as a good training data source for researches that explore the application of NLP in the Legal Domain.

1.2 Word and Sentence Embeddings

NLP consists of many techniques such as parts of speech tagging, sentiment analysis, text generation and language translation among many others. Regardless, unstructured data requires numerical representation to be analysed using ML techniques. For this transformation, often times, the text data is converted in to vector format or in other words, embeddings in order to be processed using machine learning and deep learning techniques. There are a lot of state of the art word embeddings such as Word2Vec (Mikolov et al., 2013a), Glove (Pennington et al., 2014), FastText(Mikolov et al., 2018), BERT (Devlin et al., 2018), and RoBERTa (Liu et al., 2019) and sentence embeddings available today such as Sentence-BERT (SBERT) (Reimers and Gurevych, 2019),Universal Sentence Encoder (USE) (Cer et al., 2018) and InferSent (Conneau et al., 2017). The main draw-

back in directly using these embeddings and approaches is that they have been designed and evaluated for general purpose datasets and applications. In addition the datasets used in these approaches are general purpose corpora. These embeddings are very useful for domain independent tasks but may perform poorly in domain dependant tasks.

1.3 Domain specific embeddings

The legal domain has rare vocabulary terms such as *"Habeas Corpus"*, that are rarely found in domain independent corpora. In addition, some common words imply a context specific meaning in the legal domain. For example the word *"Corpus"* in *"Habeas Corpus"* and *"Text Corpus"* gives different meanings in the legal domain. This aspect is also not captured when training with general corpora. Also approaches that are specifically designed for the legal domain should be researched to address the inherent complexities of the domain. Considering all these facts, this paper discusses the designing and training of a legal domain specific sentence embedding based on criminal sentence corpora.

2 Related Work

A lot of ground-breaking researches have been conducted in the past years, contributing to the evolution of NLP. This research makes use of a lot of these researches for both intuition and auxiliary purposes, which are discussed in this section. It is important to highlight that most of the sentence embeddings that have demonstrated state of the art performances, have been trained with large annotated datasets as discussed in subsection2.2.

2.1 Word Embeddings

The targeted word embedding of Word2Vec (Mikolov et al., 2013a) is in the continuous vector format. They have considered the facts, that Latent Semantic Analysis (LSA) is poor at preserving linear regularities of embeddings, and the computational demand of Latent Dirichlet Allocation (LDA) with large datasets. The Word2Vec architecture is a 2 layer neural network which uses two techniques, 1) Continuous Bag of Words (CBOW) and 2) Skip-gram. Words that appear in a similar context is assumed to have a similar meaning. CBOW is preferred for small corpora and faster training whereas Skip-gram performs better with large corpus but trains slower. Also in a later publication (Mikolov et al., 2013b) they propose several improvements. One improvement is the sub sampling of frequent words which reduces the training time preserves rare words and increases their accuracy. Also, negative sampling is introduced where set of words with incorrect label is used. The difference of phrases in contrast to individual meanings of words, is also addressed by allowing them to be individual tokens.

Since the emergence of multi-headed attention (Vaswani et al., 2017) with transformer architectures, BERT (Devlin et al., 2018) came up with language modeling techniques to generate word embeddings. BERT is designed in a way that it can be fine tuned for a specific task with minimum changes to the architecture, unlike the other word embeddings. Since BERT uses sub words, out of vocabulary words can be also embedded easily which is an important aspect, but they may not be as accurate when originally trained on.

As an advancement to the Masked Language Modeling (MLM) proposed by BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019) introduces many improvements. They have experimented on the impact of tuning several hyper-parameters and optimization algorithms as well as the training data. They have identified the Next Sentence Prediction (NSP) task is not improving accuracy and have only used MLM. Also in contrast to static masking in BERT (Devlin et al., 2018) they have identified dynamic masking improves accuracy. Finally by increasing batch size, adding training data and pre-training longer have achieved better performance than existing BERT, XLnet (Yang et al., 2019) models in many evaluation tasks.

XLnet (Yang et al., 2019) is developed with the intention of benefiting from both aspects of 1) Auto-regressive models that use Long Short Term Memory (LSTM) and 2)Auto-encoding models such as in BERT (Devlin et al., 2018). Also it is designed to be used for mainstream NLP tasks. In order to combine the two approaches they use the auto-regressive nature of referring to only the context seen before, and use a generated permutation of the input to give access to the whole context. With these improvements they have been able to beat BERT (Devlin et al., 2018) at many mainstream NLP evaluations.

Glove (Pennington et al., 2014) takes into con-

sideration the drawbacks of the existing model families 1) Matrix factorization models and 2) Context window related models. The context used in Glove is derived from a window where the approximation of similarity for the two word pairs considered at a time, is inversely proportionate to the distance between the two words. Glove has been trained on comparatively a large amount of data than other existing methods but uses fewer dimensions, and has been able to beat the performance of them at many evaluation tasks.

FastText (Mikolov et al., 2018) is designed to account for morphology of words, where words can take different forms which is not captured in models like Word2Vec(Mikolov et al., 2013a). FastText does this by using N-grams as the tokens, which leaves provision for out of vocabulary or misspelled words. They have achieved better accuracy with significant drop in training time.

2.2 Sentence Embeddings

SBERT (Reimers and Gurevych, 2019) uses two BERT (Devlin et al., 2018) encoders to encode the words in two sentences. The vectors of the words are then pooled using mean pooling to get a singe vector for each sentence. These vectors can then be passed on to a Soft-max classifier for classification or cosine similarity function for regression. The Stanford Natural Language Inference (SNLI) dataset is used to train this model.

Universal Sentence Encoder (USE) (Cer et al., 2018) describes two approaches for sentence embeddings, 1) transformer base models and 2) Deep Averaging Networks (DAN). The transformer based approach is high in complexity and consumes more resources but it is more accurate. In contrast DAN based models give less accuracy with less resource consumption. Multiple downstream tasks are used to make the model more generalized. Out of the two approaches, they have concluded that overall, the transformer based approach is better in accuracy.

Unlike the common unsupervised approaches, InferSent (Conneau et al., 2017) model is a sentence embedding trained with a supervised approach. Initially they have experimented with several architectures with techniques such as LSTMs and Gated Recurrent Units (GRU) and different pooling techniques. Since the approach is supervised, researchers have used the SNLI dataset. They have demonstrated that using Bidirectional LSTM (Bi-LSTM) and max pooling along with a supervised approach can outperform existing unsupervised approaches.

Researchers (Wieting et al., 2015), have formalized a way to obtain sentence embeddings considering the paraphrastic nature of sentence pairs by calculating cosine similarities of embeddings. Initially they have experimented with many approaches, with the simplest being averaging word vectors, to LSTMs. They have identified that averaging word vectors is outperforming LSTMs at many tasks, but LSTMs are better at sentiment classification tasks.

The requirement for a baseline to evaluate sentence embeddings, is addressed by researchers (Arora et al., 2017) which is mainly motivated by the work of (Wieting et al., 2015) They have stated that it can be used to evaluate domain specific sentence embeddings. Compared to the (Wieting et al., 2015) approach, they have identified that, rather than using a simple averaging function, smoothing inverse frequency techniques perform better, even more so than some LSTM and RNN approaches. Similar to the research (Wieting et al., 2015) they have identified LSTMs and RNNs are much capable of sentiment related tasks.

Sent2Vec (Pagliardini et al., 2018) have identified that all language representational learning approaches have either complex deep models trained on large datasets with expensive computing or Matrix factorization methods which is less computationally expensive but effective on large corpora. Similar to "Towards universal paraphrastic sentence embeddings" (Wieting et al., 2015), researches have also considered that mean pooling of word embedding have outperformed complex models with LSTMs. Therefore they have explored the aspect of achieving higher accuracy with less complex models. They have extended the CBOW approach in Wor2Vec (Mikolov et al., 2013a) while introducing dynamic window and n-grams.Their main contribution is a model less complex, efficient and scalable and at the same time higher in performance.

2.3 Encoder-Decoder Models

The work of (Luong et al., 2015) elaborates the usage of Encoder-Decoder architecture based on RNNs for Machine Translation tasks. They have incorporated an attention mechanism on the output sequence of the Encoder to iteratively decode

translated text for the given input. In another study (Datta et al., 2020), authors have used Recurrent Neural Networks (RNNs) in an encoder-decoder structure for neural machine translation. They have conducted the study for translating English into French Language.

2.4 STS Dataset

Semantic Textual Similarity (STS) is a measurement used to assess the closeness of two text content with respect to their semantic meaning. An evaluation toolkit for universal sentence representation is defined which makes use of STS dataset (Conneau and Kiela, 2018). STS Benchmark dataset (Cer et al., 2017) consists of pairs of sentences and the corresponding STS Scores manually annotated for each pair of sentences. STS score is a value between 0 and 5 where the perfect semantic similarity between two sentences is represented by the score of 5. Scores close to 5 represents the sentence pairs that are somewhat producing the same meaning while scores close to 0 represents irrelevant sentence pairs.

This dataset is used for training state-of-the-art sentence embeddings by (Reimers and Gurevych, 2019) through supervised learning approaches. They have shown that sentence embeddings trained using supervised learning perform significantly better in semantic text comparison tasks compared to sentence embeddings trained using unsupervised or self-supervised methods. The supervised training approach used by (Reimers and Gurevych, 2019) optimizes the model based on the difference between the cosine similarity of a sentence pair and the normalized STS score (within the range 0 to 1). The goal of this optimization is to move the vectors representing semantically similar sentences close in the high-dimensional vector space. Moreover, correlation results generated by evaluating models for STS Benchmark dataset is used in comparing the performance of sentence embeddings in general (Reimers and Gurevych, 2019; Huang et al., 2021).

3 Methodology

In this section, the data extraction process, preprocessing steps, word embedding training and sentence embedding training phases are discussed.

3.1 Dataset

The dataset used for the training of the embedding was extracted from the United States Supreme Court Case Law records extracted from FindLaw website[1]. The case law documents were chosen from Criminal Cases ranging from the year 2000 to 2010.

3.2 Pre-processing

Initially, the text files containing extracted court cases were processed to filter the body of the texts by removing title and footnotes sections in the documents. Stanford NLP python library: Stanza (Qi et al., 2020) is used to split sentences from the case texts. After observing some anomalies in split sentences, following pre-processing steps are applied to case paragraphs.

- Replaced abbreviations specific to legal domain with their long form

  Fed.R.Crim.P. – Federal Rule of Criminal Procedure

  Fed.R.Evid. – Federal Rule of Evidence

  Fed.R.Civ.P. - Federal Rule of Civil Procedure

- Removed non-ascii characters

- Removed content within rounded brackets if there are more than 2 words

  contained references and citations for legal documents

  no semantic meaning with respect to containing sentence

Following text pre-processing methods are applied to case sentences to make the text compatible for tokenization.

- Removed square brackets around letters and words

  Ex: [T]he, [petitioner], refer[s]

  Reason: caused due to styles used in web pages

- Removed numbering from the start of topic sentences

  Ex: I., A., II., 1.

[1] https://caselaw.findlaw.com/

- Replaced citations of previous cases with [CITE] keyword

  Ex: Pennsylvania v. Muniz, 496 U.S. 582, 601, 110 S.Ct. 2638

  Reason: reduce the distortion caused by citations for the semantic meaning of the sentence

- Removed sentences with more than 25% of [CITE] keyword with respect to all words

- Replaced continuous dashes, commas, white spaces with single entities

3.3 Word Embeddings for Legal Domain

Text corpus of 10,000 cases (extracted in section 3.1) containing more than 3 million words is used to train word embeddings. 300-dimensional Vectors are trained for 54059 unique words which appear more than 2 times within the corpus. Word2Vec (Mikolov et al., 2013a) and FastText (Mikolov et al., 2018) models are trained using Gensim library [2] and GloVe (Pennington et al., 2014) model is trained using glove_python [3] library. Window of 5 tokens before and after a token is used to specify the context when training the models.

3.4 Auto-Encoder Model

Due to the availability of a large in-domain text dataset, we searched for an unsupervised approach for learning sentence embeddings for the legal domain. We came up with the Auto-Encoder architecture, inspired by the application of Encoder-Decoder architecture in Neural Machine Translation systems (Datta et al., 2020; Luong et al., 2015). The objective of the Auto-Encoder is to reconstruct the original sentence token-by-token in an iterative manner using the state from previous tokens of the sentence and the vector representation for the whole sentence generated by the Encoder.

The workflow of the Auto-Encoder for a sentence containing m tokens at the $(k\text{-}1)^{th}$ iteration of the decoder is displayed in Figure 1. Upper section of the diagram represents the Encoder and lower section, the Decoder. The Embedding layers used

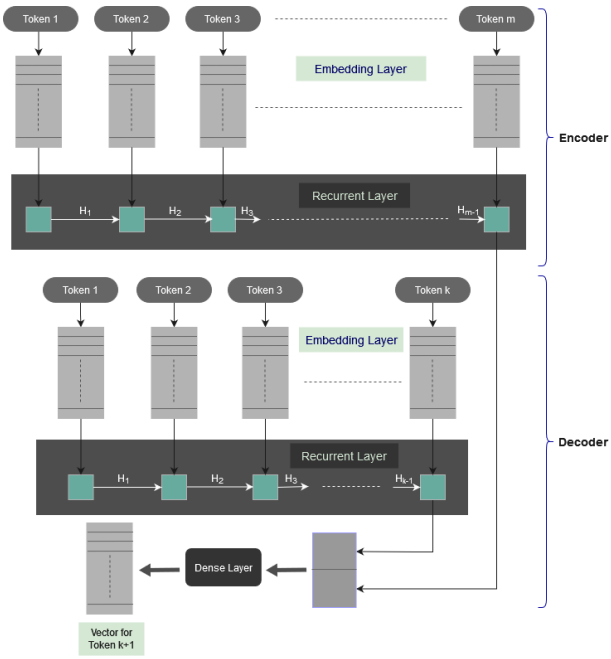

Figure 1: Auto-Encoder Architecture

in both Encoder and Decoder share same embedding matrix populated with pre-trained word embeddings.

Encoder takes in a sentence as a sequence of tokens and outputs a vector representation for the sentence. According to Figure 1, Embedding layer outputs a sequence of m vectors which is then passed on to a Recurrent layer with a specified number of units. The final state vector of the Recurrent layer is considered as the sentence embedding which is passed on to the decoder.

Each sentence is padded from the beginning with a *[START]* token and a *[END]* token to mark the beginning and the end of a sentence. Decoder iteratively predicts the next token starting from the *[START]* token at the first iteration to predict the token after the *[START]* token. Figure 1 depicts the $(k\text{-}1)^{th}$ iteration of the decoder, where the vector for $(k+1)^{th}$ token is predicted. Decoder takes in the k tokens preceding the $(k+1)^{th}$ token and passes them to the Embedding layer which outputs a sequence of k vectors. These vectors are passes on to a Recurrent layer where the final state vector is concatenated with the sentence vector provided by the Encoder. These concatenated output is passed on to a Dense layer which outputs a vector with the same dimension of the pre-trained embeddings.

Training loss is calculated at each decoding iteration, using the mean squared error between the predicted token vector and the actual vector ob-

[2] https://radimrehurek.com/gensim/

[3] https://github.com/maciejkula/glove-python/

tained from pre-trained word embeddings. Cosine similarity is used as the accuracy metric to evaluate the similarity between predicted and pre-trained word vectors.

The ability to predict the next token at each decoding step is based on the semantic meaning captured by the Encoder for the complete sentence and the state captured by the Decoder's Recurrent layer about the tokens preceding the to-be-predicted token of the sequence.

3.5 STS Dataset for Legal Domain

Understanding the need of a labeled dataset for legal domain to train and evaluate sentence embeddings. An STS dataset was prepared using sentences taken from US Supreme Court criminal cases and combining legal specific sentence pairs taken from STS Benchmark dataset with the assistance of a legal professional. Sample sentence pairs taken from the prepared dataset is displayed in Table 1.

Table 1: STS Legal Dataset - Samples

Sentence 1	**Sentence 2**	**Score**
Petitioner explained that his actions were taken in self-defense.	During the court proceedings, plaintiff argued he was only trying to save himself.	4.25
He did not present any evidence.	There were no evidence to support him.	3.25
The memorandum argued that plaintiff was not a risk to public safety and that he had accepted responsibility for his crime.	Court hearing raised concerns about the public safety.	1.5

First pair of sentences in Table 1 has a high STS score because the semantic meaning is same despite some content before the second sentence. Second pair of sentences has somewhat lower score since the first sentence doesn't elaborate the need for the petitioner to present evidence. It could be about supporting himself or against the opponent party. Last sentence pair has a low score since they are irrelevant despite the mention of public safety.

3.6 Multi-task Model for learning Sentence Embeddings

Since we have prepared a large dataset of case sentences and obtained a labeled dataset of STS score annotated sentence pairs, we focused on training a model for multiple tasks. To make use of the large set of unlabeled sentences, we defined a task to determine whether a sentence is distorted or not. Legal STS dataset is used for the task of predicting the similarity between two sentences and evaluating against the STS score. We list down the two tasks that the model is trained for:

- Noise added sentence discrimination
- Semantic similarity between a sentence pair

Noise addition process for sentences is done using a random word replacement algorithm. First, a set of general english words is extracted from the case sentence dataset. This set of words does not contain any person names, organization names or punctuation marks. Total number of general words accounts for 17796.

This set of words is used to replace 20% of words within each sentence by picking randomly. With this word replacement, the semantic meaning of the sentence is distorted. An example is displayed in Table 2.

Table 2: Noise Addition for Sentences

Original Sentence	**Distorted Sentence**
Plaintiff argued that the district court decision was unreasonable.	Plaintiff **guilty** that the district court **an** was unreasonable.

50% of the sentence dataset is distorted by random replacement and the label 1 is assigned for each distorted sentence. Label 0 is assigned for each original sentence.

According to Fig. 2, The model is trained for sentence discrimination task and sentence pair similarity task at each training step. Model shares the same embedding layer and Recurrent Neural Network (RNN) layer for both tasks. Sentence Dataset provides a batch of sentences containing original and distorted sentences and from the Dense layer output, the probability of a sentence being either original or distorted is calculated. Discrimination loss is then calculated using

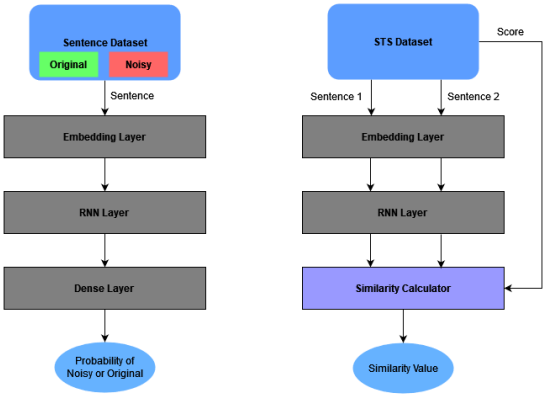

Figure 2: Multi-task model Architecture

this probability and the actual label. At the same training step, STS dataset provided a batch of sentence pairs and the similarity calculator produces the cosine similarity between the two vectors output by RNN layer. STS loss is calculated using the similarity value and the STS score provided by the dataset. Model weights are optimized using both Discrimination loss and STS loss.

This multi-task approach aims to train the model to capture the semantics of the sentences while preventing the model from over-fitting for STS Dataset which is relatively small compared to Sentence Dataset. Discrimination task force the model to identify distortions only by looking at the sentence vector provided by the RNN layer. STS task trains the model to move vectors of similar sentences closer in the vector space and irrelevant sentences further away. This approach is suitable for a setting where a large corpus of unlabeled data is available along with a small set of labeled data and the annotation cost is high to expand the labeled dataset.

4 Experiments and Results

In this section we discuss the variations that were done to identify most effective configurations. Several variations were experimented with the choice of word embeddings trained on general corpus and legal domain corpus. Also few variations were also tested with the auto encoder model to identify relatively better combination.

4.1 Pre-trained Word Embeddings

For our experiments, 3 types of word embeddings trained on general corpora and the legal corpus of 10,000 US Supreme Court cases, are used to initialize the Embedding Layer of the Auto-Encoder model. Pre-trained word embeddings on general corpora are obtained from online sources.

- Word2Vec
- Glove
- FastText

In the token distribution for 10,000 cases, 52% of the total sentences contain tokens within the range 20 - 40. We will be referring to the word embedding types trained on this legal corpus as *Word2Vec_Legal*, *GloVe_Legal* and *FastText_Legal* for the purpose of distinguishing them from word embeddings trained on general corpora.

4.2 Auto-Encoder Results

1000 US Supreme court cases consisting of 125719 sentences are used for the training and evaluation of the Auto-Encoder model. Experiments are done based on the word embedding type and Recurrent layer type. All the variations listed in Table 3 are trained for 20 epochs and measured the results as a controlled experiment to choose the relatively best variation for further training.

4.3 Multi-task Model Results

Multi-task Model of Noise Discrimination and STS tasks is trained and evaluated using different configurations of GRU layers. Accuracy and F1 scores are recorded for Noise Discrimination task and STS evaluation results are recorded using Pearson and Spearman Correlation between predicted similarity value and the STS score. Table 4 displays the results.

5 Conclusion and Future Work

Despite the availability of massive amount of text data, legal domain has inherent domain complexities and suffers from lack of annotated data. In this research we have conducted experiments with several variations to identify suitable sentence embedding models for the legal domain with this low resource settings. Self supervised approach is leveraged to overcome the lack of annotated data in the domain. This research serves as a preliminary step towards getting a proper numerical representation of a legal case. We intend to use the insights gained from this research to advance the sentence embeddings for more accurate results, with the use of relatively higher computational resources effectively.

Table 3: Model Variation Metrics

RNN Type	Units	Word Embedding	Train Cosine Sim.	Validation Cosine Sim.
GRU	512	Glove	0.2663	0.2578
		Word2Vec	0.2068	0.2033
		FastText	0.3323	0.3299
		Glove Legal	0.2776	0.2743
		Word2Vec Legal	0.2358	0.2310
		FastText Legal	0.2182	0.2153
Bi-GRU	512	Glove Legal	0.2550	0.2499
		Word2Vec Legal	0.2249	0.2180
		FastText Legal	0.2139	0.2097

Table 4: Multi-task Model Metrics

RNN Type	**Units**	**Accuracy**	**F1 (Original)**	**F1 (Noisy)**	**Pearson C.**	**Spearman C.**
GRU	512	92.21	91.94	92.46	46.81	48.09
GRU	768	93.70	93.71	93.70	57.62	51.34
LSTM	512	89.68	89.36	89.98	39.93	35.25
LSTM	768	92.73	92.52	92.92	34.35	28.76

We intend to make use of the derived sentence embedding models to legal domain specific tasks such as winning party prediction of legal cases.

References

[Arora et al.2017] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*.

[Cer et al.2017] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

[Cer et al.2018] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

[Conneau and Kiela2018] Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

[Conneau et al.2017] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September. Association for Computational Linguistics.

[cor2020] 2020. Case law. https://www.law.cornell.edu/wex/case_law. Accessed: 2021-05-27.

[Datta et al.2020] Debajit Datta, Preetha Evangeline David, Dhruv Mittal, and Anukriti Jain. 2020. Neural machine translation using recurrent neural network. *International Journal of Engineering and Advanced Technology*, 9(4):1395–1400.

[Devlin et al.2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Huang et al.2021] Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. Whiteningbert: An easy unsupervised sentence embedding approach. *arXiv preprint arXiv:2104.01767*.

[Liu et al.2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly opti-

mized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[Luong et al.2015] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

[Mikolov et al.2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mikolov et al.2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[Mikolov et al.2018] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

[Pagliardini et al.2018] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In *NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics*.

[Pennington et al.2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[Qi et al.2020] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

[Reimers and Gurevych2019] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11.

[Vaswani et al.2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

[Wieting et al.2015] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.

[Yang et al.2019] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.