Contract Review Automation with Attention Mechanism

Kenwar, Sanjeev
skenwar3@gatech.edu

Kanuri, Sekhar
cvenkata6@gatech.edu

Ramakrishna, Shravan
sramakrishna7@gatech.edu

Gudiduri, Siddharth
sgudiduri3@gatech.edu

Abstract

The automation of contract reviews via contract level natural language inference (ContractNLI) will be a value-adding endeavor given the plethora of written agreements that govern the majority of all business-to-business transactions [18]. Taking into consideration the associated costs involved in facilitating manual reviews, and given the risk of smaller companies and individuals putting themselves at a disadvantage by shunning such reviews altogether due to dearth of resources, there exists plenty of scope and opportunity to build a model that automates this. There is existing work [6] in this area and we seek to enhance the SOTA model [3] in an attempt to improve classification performance metrics. To this end, we implement a decomposable attention model [9] that uses the attention mechanism. We also use focal loss [2] to mitigate class imbalance and focus learning on hard misclassified examples. Facebook AI Similarity Search (Faiss) [15] is introduced in the context of challenging the model by sampling neutral cases that resemble entailment. In this report, we describe our experimental setup, document our experiences and challenges, and report our results from the implementation.

1. Motivation

We embarked on this exercise using a decomposable attention model [9] that attempts to identify if the content of a given contract document agrees, disagrees or is neutral with one of 17 different questions asked of it by a user. Koreeda and Manning [6] view these questions as hypotheses, and the contents of the contract as premises. Thus, we aim to determine the logical relationship between a pair of text sequences, and these relationships fall into the following three categories below. Examples included are text sequences obtained from the Stanford ContractNLI dataset.

- **Entailment**: The hypothesis can be inferred from the premise. Example -
  - *Premise* - all confidential information in any form and any medium including all copies thereof disclosed to the recipient shall be returned or destroyed
  - *Hypothesis* - receiving party shall destroy or return some confidential information upon the termination of agreement
- **Contradiction**: The negation of the hypothesis can be inferred from the premise. Example -
  - *Premise* - {d}. erase and or destroy any confidential information contained in computer memory or data storage apparatus under control of or used by mentor
  - *Hypothesis* - receiving party may retain some confidential information even after the return or destruction of confidential information
- **Neutral**: All the other cases. Example -
  - *Premise* - any other party with the discloser's prior written consent and
  - *Hypothesis* - receiving party shall not reverse engineer any objects which embody disclosing party's confidential information

The Attention mechanism [1] was initially proposed as an approach to address machine translation. Attention Is All You Need [26] expanded on this concept and established the Transformer architecture using multiplicative attention. BERT [3] pre-trained models that came later added fine-tuning capabilities with just one additional output layer to create SOTA models for a wide range of tasks. Koreeda and Manning [6] annotate and release a dataset containing 607 contracts and introduce Span NLI BERT [7], a multi-task Transformer model that can jointly solve NLI and evidence identification. One limitation of using the NLI BERT model is the number of parameters that can be tuned, which runs to approximately 110M for the base

version, and makes the training computationally expensive. By aligning the substructures of the text sequence and aggregating the information, we can build a model with fewer parameters that we hypothesize will perform almost as good as SOTA.

Businesses and individuals care about the time and capital expended in the process of manual contract reviews, and therefore will immensely benefit from automation of the process. We deem the application of Faiss to the contract review algorithm as being novel as it enables the model to better discriminate between entailment cases and closely resembling neutral cases. The application of focal loss is also novel as SOTA models don't consider this. Success in this endeavour would result in reduction of space complexity and a more efficient use of computational resources. Space complexity would be significantly smaller due to the smaller set of parameters in the decomposable attention model [9] as compared to the SOTA model (BERT).

ContractNLI [20] is a dataset for document-level natural language inference (NLI) on contracts, and the goal here is to automate/support the time-consuming procedure of contract review. The train data contains 423 documents, while the test data contains 123 documents. The structure of a given document is shown in [1]. At the document level, we have a "text" key that contains the full document text, and the "spans" key that splits the text into a bunch of premises. The key "annotation_sets" is a list that contains multiple annotations for a given document. At the annotation level, every key "nda-1", "nda-2", etc. is a hypothesis which either entails, contradicts, or is neutral to the given document. The "spans" key under each hypothesis is used to index the "spans" key at the document level. Example - "nda-1" entails the spans 1, 13, and 91. Here, span 1 at the document level corresponds to sentence text indexed between characters [25, 89]. The "labels" key describes the text sequence for each hypothesis.

```
{
  "documents": [
    {
      "id": 1,
      "file_name": "example.pdf",
      "text": "NON-DISCLOSURE AGREEMENT\nThis NON
          -DISCLOSURE AGREEMENT (\"Agreement\")
          is entered into this ...",
      "document_type": "search-pdf",
      "url": "https://examplecontract.com/example
          .pdf",
      "spans": [
        [0, 24],
        [25, 89],
        ...
      ],
      "annotation_sets": [
        {
          "annotations": {
            "nda-1": {
              "choice": "Entailment",
              "spans": [
                1,
                13,
                91
              ]
            },
            "nda-2": {
              "choice": "NotMentioned",
              "spans": []
            },
            ...
          }
        }
      ]
    },
    ...
  ],
  "labels": {
    "nda-1": {
      "short_description": "Explicit
          identification",
      "hypothesis": "All Confidential Information
           shall be expressly identified by the
          Disclosing Party."
    },
    ...
  }
}
```

Listing 1. Dataset Structure

2. Approach

Our initial approach involved a naive method of identifying neutral premises that are in close proximity to entailment premises using Euclidean distance. We observed that this was an extremely computationally expensive process due to the large number of premises involved. We later discovered that Faiss remedies this issue and augments the training process by introducing Voronoi cell based clustering and product quantization (PQ) [2.1].

We had anticipated the following issues -

- The inclusion of Faiss-selected neutral cases would challenge the model by making it work harder to discern between entailment and neutral cases.
- Remnants of legal contract-specific parlance despite data pre-processing techniques such as tokenization, stemming, and lemmatization, which may affect model classification.

2.1. Facebook AI Similarity Search (Faiss)

Faiss is a library optimized for memory usage and speed, and it allows rapid searches for multimedia documents

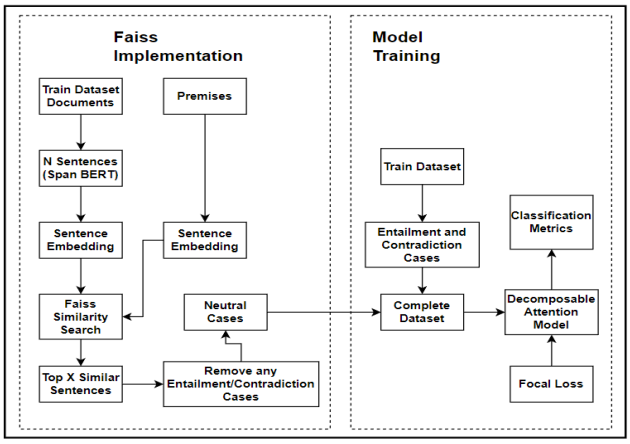

Figure 1. Training Flowchart

similar to a document in question, by employing traditional machine learning techniques such as K nearest neighbors (KNN) and K-means clustering [4].

From [1], we can observe that we initially begin with a train set of documents and the hypotheses. We perform sentence level embedding using a $BERT_{BASE}$ transformer [11] that maps a sentence to an embedding of dimension 768. The BERT model is pre-trained using Next Sentence Prediction (NSP) [27], which attempts to establish relationships between sentences, and is carried out by randomizing the next sentence 50% of the time. This makes the output conducive to Faiss.

$$f_d = argmin \sqrt{\sum_{i=1}^{d} (v_e^{(i)} - v_n^{(i)})^2} \ni v_e^{(i)}, v_n^{(i)} \in \mathbb{R}^d \quad (1)$$

where v_e and v_n refer to entailment and neutral premise vectors and f_d refers to the distance between the vectors.

Before feeding the premises to Faiss, we prepare an index using the train set of documents. This limits the computation to the distances between vectors. By obtaining the top $X \in \mathbb{Z}^+$ similar neutral premises for a given premise, and eliminating entailment and contradiction cases, we are left with neutral cases that we provide as input to the model, along with the original entailment and contradiction cases obtained from our train dataset.

Faiss implicitly partitions the corpus into several Voronoi [24] cells with centroids in order to make the search faster. The distance between the query vector and each of the centroids is computed, and vectors from the specific Voronoi cell pertaining to the closest centroid are chosen to be compared with. This speeds up the search by avoiding comparison with sentence vectors in the entire corpus.

Product Quantization is used to improve speed at the cost of accuracy and involves splitting our original embedding vectors into sub-vectors. For each set of sub-vectors, clustering is performed to create multiple centroids for each set. Each sub-vector is then replaced with an identifier to its nearest set-specific centroid.

Product Quantization speed improvements come at the cost of accuracy. Due to the nature of our problem, we can afford to include PQ to improve on speed as we sample X (10 for our use case) neutral cases.

2.2. Decomposable Attention Model

The following sections describe the parts of the Decomposable Attention Model [2.2] and how they are put together. This pertains to the Model Training Section in [1].

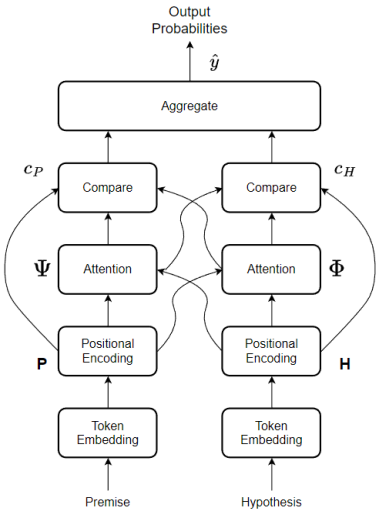

Figure 2. Decomposable Attention Model

2.2.1 Input Representation

Consider Premise P denoted by $(p_1, p_2, ..., p_a)$ and Hypothesis H denoted by $(h_1, h_2, ..., h_b)$ where a, b refer to the number of tokens (sequence length) in $p_i, h_j \in \mathbb{R}^d$ respectively, where d is the word embedding dimension.

Global Vectors (GloVe) [10] for word representation is an unsupervised learning algorithm for obtaining vector representations for words, with training performed on aggregated global word-word co-occurrence statistics from a corpus. To perform word embedding, we use GloVe 6B 100d.

Positional Encoding [2, 3] uniquely encodes information about the position of a token.

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \quad (2)$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \quad (3)$$

Here, d is the embedding dimension, pos is the position of the token in the sequence, and i maps to sin and cosine functions.

2.2.2 Attending

We perform soft alignment of the premise and hypothesis using [1]. This is essentially achieved by passing the input premise and hypothesis through a multi-layer perceptron and then computing soft attention weights [16], [28] using [4]

$$w_{ij} = F(p_i)^T F(h_j) \quad (4)$$

where F is the multi-layer perceptron with ReLU non linear activation that maps p_i, h_j to a hidden dimension space. This allows us to calculate the projection of the premise over the hypothesis [4].

In [5], we obtain a representation of the hypotheses $(h_1, h_2, ..., h_b)$ softly aligned with premise p_i. The hypotheses are weighted by a Softmax activation function for a given premise and vice versa [6]. The intuition behind the *alignment model* is based on a bidirectional RNN used as an encoder and decoder [1].

$$\Phi_i = \sum_{j=1}^{b} \frac{exp(w_{ij})}{\sum_{k=1}^{b} exp(w_{ik})} h_j \quad (5)$$

$$\Psi_j = \sum_{i=1}^{a} \frac{exp(w_{ij})}{\sum_{k=1}^{a} exp(w_{kj})} p_i \quad (6)$$

2.2.3 Comparing

In the compare section, all the tokens from one sequence, with their corresponding weights are compared with a token in the other sequence.

$$c_{P,i} = G([p_i, \Phi_i]), i = 1...a \quad (7)$$

$$c_{H,j} = G([h_j, \Psi_j]), j = 1...b \quad (8)$$

The representation $c_{P,i}$ [7] is the concatenation of premise token p_i and the softly aligned weight representation for that token Φ_i. A similar operation is performed for the hypothesis as well [8]. As the concatenation operation is performed along the embedding dimension, the multi-layer perceptron G maps input dimension equal to twice the embedding dimension, to the number of hidden units.

2.2.4 Aggregating

The final step performed by the decomposable attention model is aggregating the information obtained from the comparison step. The information in the comparison vectors are aggregated through a summation operation [9, 10]. The summed up results are now fed into a multi-layer perceptron H and are mapped to the number of outputs - Entailment, Contradiction and Neutral [11].

Thus, the learnable parameters in the model refer to the functions F, G and H in equations [4, 7, 8, 11].

$$c_P = \sum_{i=1}^{a} c_{P,i} \quad (9)$$

$$c_H = \sum_{j=1}^{b} c_{H,j} \quad (10)$$

$$\hat{y} = H([c_P, c_H]) \quad (11)$$

2.3. Focal Loss

While training the decomposable attention model, we use focal loss [2] as there exists class imbalance among the 3 classes - Entailment, Contradiction and Neutral.

$$CB_{focal}(z, y) = -\frac{1-\beta}{1-\beta^{n_y}} \sum_{i=1}^{C} (1-p_i^t)^{\gamma} log(p_i^t) \quad (12)$$

The β hyper-parameter can be tuned to perform re-weighting. When p^t is small and consequently, $(1-p^t)^{\gamma}$ is close to 1, then [12] becomes classic cross entropy, and would result in incorrect classification by the model. As the model adjusts its weights [4, 7, 8, 11], it scales down the contribution of easy examples during training and instead focuses on the harder examples, resulting in an improvement in prediction accuracy for the minor classes.

3. Experiments and Results

We measure success by total accuracy and F1 scores for the individual classes. We consider two sets of Experiments.

1. Training the model on the standard ContractNLI Dataset
2. Training the model on the ContractNLI Dataset where neutral cases are sampled through Faiss.

Post pre-processing, we obtain the following number of records for the train and test dataset -

Train Dataset - Number of Records		
Entailment	Contradiction	Neutral
6759	1578	33711

Test Dataset - Number of Records		
Entailment	Contradiction	Neutral
1944	472	9720

3.1. Standard Contract NLI Dataset

In order to curate this dataset, we obtain the Entailment and Contradiction cases as-is from the dataset. For the neutral premises however, we sample a small arbitrary and configurable set of 10 neutral premises for each hypothesis for a given document.

From the results in Table 1 [3.1] and Table 2 [3.1], The implementation of focal loss with β of 0.9 improves the F1 score for the Entailment class from 0.08 to 0.46. There is also a simultaneous improvement in the F1 score for the Neutral class. In the process of correctly classifying Entailment premises (which were previously incorrectly classified as Neutral), an improvement in the F1 score for the Neutral class is observed. However, this comes at the cost of an albeit small decrease in the F1 score of the Contradiction class, despite a significant improvement for the Entailment and Neutral classes. We also configure γ to 2 in order to down weight the contribution of the easily-classifiable class to the overall loss.

While training the model towards an overall lower loss level on the test set, we ensured that the high accuracy isn't skewed towards any particular class. It is worth noting here that the SOTA model only considers the accuracy and F1 score with respect to Entailment and Contradiction cases only. We additionally accounted for the F1 score of the Neutral class, especially because an empirical examination of legal contracts demonstrates that the Neutral class is almost always the dominant one.

The grid search we performed covered both Adam [5] as well as Stochastic Gradient Descent (SGD) [19] optimization techniques. We observed that SGD consistently under performs Adam, with all other hyper parameters held constant. Adam combines techniques from both Adagrad [23] and RMSprop [22], thus maintaining the first and second momentum stats for gradients. In comparison to Adam, SGD evinced over-fitting on the training data and relatively poor performance on the test set.

3.2. Contract NLI Dataset with Faiss

The incorporation of Faiss [1] removes the randomness in the sampling of Neutral premises. Instead, it provides structure to that process, picking Neutral premises that most closely resemble the Entailment premise in question. From the results in Table 3 [3.2] and Table 4 [3.2], we observe that Faiss poses a tougher problem to the model, as it needs to train harder to accurately discern between Entailment and Neutral premises. This also affects the F1 scores of the Contradiction class as most contradictions are now being predicted as Neutral.

When Faiss is combined with the use of focal loss [2], an albeit small improvement is observed in the F1 scores of the Entailment and the Contradiction classes, but at the expense of the overall accuracy as well as the F1 score of the Neutral class. This result isn't unexpected, because the model adjusts its weights in order to improve on the minority classes. In an imbalanced dataset, the majority class dominates the gradient process and loss. Thus, by using focal loss, we resolve two problems - class imbalance and classification of easy and hard examples by modifying the loss function. This is seen as a trade-off in accuracy between majority and minority classes, with a decrease in the former and an increase in the latter.

Further, in order to account for the relatively low F1 scores and the class imbalance resulting from relatively similar hypothesis-premise tuples in high dimensional space, the optimal value for β is a high 0.999. The accuracies as compared with the standard dataset are lower, and this is inline with our expectation. The smaller learning rate and batch size that we employ are reasonable given the data is essentially harder to train on, and we desire the training algorithm to take smaller steps in optimizing the objective/loss function. The Adam optimizer outperforms SGD in this scenario also.

4. Conclusion and Future Work

In this work, we applied the Decomposable Attention Model to the ContractNLI Dataset and attempted to improve on space and time complexity as compared to SOTA methods. While we were able to obtain results more or less comparable to SOTA, we observed that the implementation of Faiss resulted in a drop in accuracy and class specific F1 scores when compared to the model trained on the Standard ContractNLI Dataset.

Although the application of Faiss to this problem is a novel idea, it makes model training challenging from the perspective of discerning entailment and neutral premises that are rendered closely resemblant of one another. Future

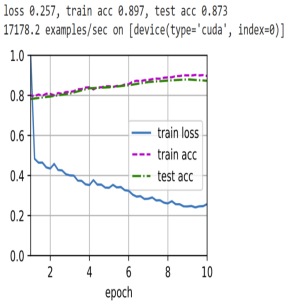

Figure 3. Loss (CE) and Accuracy Curves (Standard Dataset)

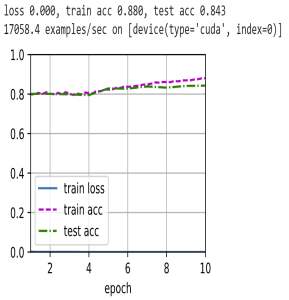

Figure 4. Loss (Focal) and Accuracy Curves (Standard Dataset)

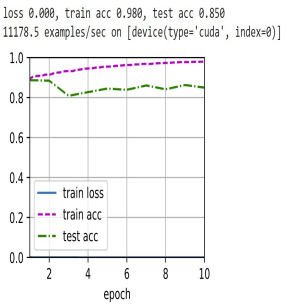

Figure 5. Loss (Focal) and Accuracy Curves (Standard Dataset with Faiss)

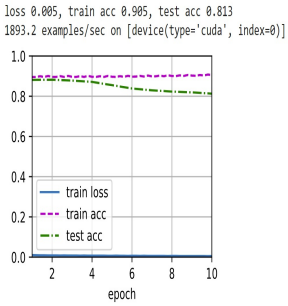

Figure 6. Loss (Focal) and Accuracy Curves (Standard Dataset with Faiss)

Standard Contract NLI Dataset - Cross Entropy Loss							
Optimizer [SGD, ADAM]	Batch Size [64, 128, 256]	Learning Rate [1e-2, 1e-3, 1e-4]	Regularization [1e-3, 1e-4, 1e-5]	Entailment F1 Score	Contradiction F1 Score	Neutral F1 Score	Accuracy
ADAM	256	1e-2	1e-3	0.08	0.28	0.54	0.88

Table 1. Optimal Hyper-parameters and Validation Results for Standard Contract NLI Dataset using Cross Entropy Loss

Standard Contract NLI Dataset - Focal Loss									
Optimizer [SGD, ADAM]	Batch Size $\mathbb{N}$ [64, 128, 256]	Learning Rate α [1e-2, 1e-3, 1e-4]	Regularization λ [1e-3, 1e-4, 1e-5]	Class Re-balance Factor β [0.75, 0.9, 0.999]	Modulating Factor γ [1,2]	Entailment F1 Score	Contradiction F1 Score	Neutral F1 Score	Accuracy
ADAM	256	1e-2	1e-3	0.9	2	0.46	0.22	0.86	0.84

Table 2. Optimal Hyper-parameters and Validation Results for Standard Contract NLI Dataset using Focal Loss

Standard Contract NLI Dataset curated using Faiss- Cross Entropy Loss							
Optimizer [SGD, ADAM]	Batch Size [64, 128, 256]	Learning Rate [1e-2, 1e-3, 1e-4]	Regularization [1e-3, 1e-4, 1e-5]	Entailment F1 Score	Contradiction F1 Score	Neutral F1 Score	Accuracy
ADAM	128	1e-3	1e-3	0.11	0.02	0.92	0.85

Table 3. Optimal Hyper-parameters and Validation Results for Standard Contract NLI Dataset using Faiss and CE Loss

Standard Contract NLI Dataset curated using Faiss - Focal Loss									
Optimizer [SGD, ADAM]	Batch Size $\mathbb{N}$ [64, 128, 256]	Learning Rate α [1e-2, 1e-3, 1e-4]	Regularization λ [1e-3, 1e-4, 1e-5]	Class Re-balance Factor β [0.75, 0.9, 0.999]	Modulating Factor γ [1,2]	Entailment F1 Score	Contradiction F1 Score	Neutral F1 Score	Accuracy
ADAM	128	1e-3	1e-3	0.999	1	0.16	0.09	0.85	0.81

Table 4. Optimal Hyper-parameters and Validation Results for Standard Contract NLI Dataset using Faiss and Focal Loss

research in this space could involve exploring variable volumes of sampled neutral cases. Future work could also involve improving on the representational aspects of sentence embedding, that when combined with Faiss and focal loss, would allow the model to better attend to the relationships between premises and hypotheses.

5. Work Division

We split up into two teams [5] of two members each, and worked together on several overlapping aspects of the project. One team comprised of Sanjeev Kenwar and Sekhar Kanuri, and the other comprised of Siddharth Gudiduri and Shravan Ramakrishna. We met twice weekly between the start of October and early December.

Student Name	Contributed Aspects and Details
Kenwar, Sanjeev	1. Project conception, vision and overall guidance 2. Data Loading and pre-processing - PyTorch DataLoader. 3. Code Templating 4. Implementation of Faiss Code and validating usability for ContractNLI dataset 5. Model training with hyper-parameters
Kanuri, Sekhar	1. Literature Review of NLI, Faiss, Word Embedding techniques (BERT, Glove, word2vec) and SPANs. 2. Implementation of Faiss Code and validating usability for ContractNLI dataset 3. Implementation of Grid-Search Code for Model Training. 4. Implementation of classification report with F1 scores, Precision, Recall and Accuracy. 5. Generation of Accuracy and Loss curves and experimentation with hyper-parameters.
Gudiduri, Siddharth	1. Literature Review of Decomposable Attention Model and Neural Machine Translation paper 2. Implementation of the Decomposable Attention Model 3. Code Modularization and YAML setup for hyper-parameter tuning 4. Implementation of Focal Loss 5. Model training with hyper-parameters
Ramakrishna, Shravan	1. Literature Review of Decomposable Attention Model and Neural Machine Translation paper 2. Implementation of the Decomposable Attention Model 3. Logical Code Structuring 4. Implementation of Focal Loss 5. Model training with hyper-parameters

Table 5. Contributions of Team Members

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014. 1, 4

[2] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-balanced loss based on effective number of samples. *CoRR*, abs/1901.05555, 2019. 1, 4, 5

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. 1

[4] Simon Jégou, Michal Drozdzal, David Vázquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. *CoRR*, abs/1611.09326, 2016. 3

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. 5

[6] Yuta Koreeda and Christopher D. Manning. Contractnli: A dataset for document-level natural language inference for contracts. *CoRR*, abs/2110.01799, 2021. 1

[7] Yuta Koreeda and Christopher D. Manning. Contractnli: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, 2021. 1

[8] T. Mikolov, W.-T Yih, and G. Zweig. Linguistic regularities in continuous space word representations. *Proceedings of NAACL-HLT*, pages 746–751, 01 2013.

[9] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *CoRR*, abs/1606.01933, 2016. 1, 2

[10] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 3

[11] Nils Reimers and Iryna Gurevych. 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 2019. 3

[12] http://acl2014.org/acl2014/W14-16/pdf/W14-1618.pdf.

[13] https://arxiv.org/pdf/1301.3781.pdf.

[14] https://dl.acm.org/doi/10.1145/1390156.1390294.

[15] https://engineering.fb.com/2017/03/29/data-infrastructure/faiss-a-library-for-efficient-similarity-search/. 1

[16] https://lilianweng.github.io/posts/2018-06-24-attention/. 4

[17] https://nlp.stanford.edu/pubs/glove.pdf.

[18] https://offers.exigent-group.com/how-gcs-can-thrive-not-just-survive. 1

[19] https://paperswithcode.com/method/sgd-with-momentum. 5

[20] https://stanfordnlp.github.io/contract-nli/. 2

[21] https://www.apporchid.com/can-todays-nlp-technology-really-understand-legal-co

[22] https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf. 5

[23] https://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf. 5

[24] https://www.pinecone.io/learn/faiss-tutorial/. 3

[25] http://www.fit.vutbr.cz/research/groups/speech/publi/2009/mikolov_ic2009_nnlm_4.pdf.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. 1

[27] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 3

[28] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021. 4