ANTS: A Framework for Retrieval of Text Segments in Unstructured Documents

Brian Chivers, Mason P. Jiang, Wonhee Lee, Amy Ng, Natalya I. Rapstine, Alex Storer

Stanford University

{bchivers,mpjiang,wolee,awn,nrapstin,astorer}@stanford.edu

Abstract

Text segmentation and extraction from unstructured documents can provide business researchers with a wealth of new information on firms and their behaviors. However, the most valuable text is often difficult to extract consistently due to substantial variations in how content can appear from document to document. Thus, the most successful way to extract this content has been through costly crowdsourcing and training of manual workers. We propose the Assisted Neural Text Segmentation (ANTS) framework to identify pertinent text in unstructured documents from a small set of labeled examples. ANTS leverages deep learning and transfer learning architectures to empower researchers to identify relevant text with minimal manual coding. Using a real world sample of accounting documents, we identify targeted sections 96% of the time using only 5 training examples.

1 Introduction

Datasets of text documents hold enormous amounts of raw information, particularly for social scientists and business researchers. An individual document contains not only declarative statements and facts, but also style, theme and sentiment information that can be used to evaluate diverse research questions.

Researchers have spent decades developing frameworks and techniques to distill text into features that are easily integrated into existing research practices. One common practice is to use vetted word lists to compute a score for a particular topic or theme. For example, Loughran and Mcdonald (2011) use a vetted word list to identify the degree of uncertain language used in financial documents, and count the number of occurrences as a proxy for the amount of prospective discussion. Dictionary approaches such as LIWC (Tausczik and Pennebaker, 2009) match words to predefined psycholinguistic categories, allowing researchers to identify broad themes including "anxiety" and "religion".

More computationally sophisticated methods such as word embeddings (Mikolov et al., 2013) and topic modeling algorithms (Blei et al., 2003) provide the capability to measure prevalence of topics within documents, as well as the relationships between words and how they may shift over time. The development of transformer models such as BERT (Bi-directional Encoder Representations from Transformers) (Devlin et al., 2019) have opened a new frontier of text processing, with models trained to categorize, summarize or answer specific questions from input text.

Despite all these advancements, valuable pieces of information remain difficult to extract or categorize in large unstructured documents. Word lists and dictionaries can fail to capture the immense variety of language that can be used to talk about a single topic. Topic modeling algorithms may not capture a specific concept in one overarching topic. Thus, to ensure maximum quality, many researchers resort to manual methods to effectively characterize the text data from their documents. One approach is to begin by identifying only the segments of interest, so that only relevant text can be utilized by subject matter experts or computational methods. To manually select these relevant subsets, researchers frequently work with undergraduates or other research assistants, or they post tasks to pools of remote workers using platforms like Amazon's Mechanical Turk (MTurk). In either method, using humans to extract specific pieces of text from large documents is costly and time consuming.

We propose a general deep learning framework to provide Assisted Neural Text Segmentation (ANTS) as a way to facilitate identification of text segments of interest for researchers. The primary goal of this general framework aims to reduce the amount of time subject matter experts must spend

Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing, pages 38 - 47
July 14, 2022 ©2022 Association for Computational Linguistics

manually coding documents or identifying and training effective research assistants. The ANTS framework has four steps:

1. Label a small handful of documents indicating the relevant section of text
2. Fine-tune a pre-trained deep transformer model (e.g., BERT) on the labeled dataset
3. Classify new text with the fine-tuned model
4. Infer the section of interest from the model's classification scores combined with domain knowledge from the research question

In this paper, we present a specific problem of extracting Human Capital Disclosure (HCD) sections from Form 10-K filings created by corporations for regulatory Securities and Exchange Commission (SEC) filings. We also illustrate a few strategies to elevate the performance of our model without annotating additional training data. Through similar means, we hope to provide a less costly and time consuming pathway for researchers to identify relevant segments of text from unstructured documents.

2 Related Work

With modern advancements in deep learning technology and the increased need for processing large text datasets, researchers have been optimizing the task of automated text segmentation. Common applications of this natural language processing (NLP) task include information retrieval (Oh et al., 2007; Nguyen et al., 2021), topic segmentation (Arnold et al., 2019; Aumiller et al., 2021), and document summarization (Chuang and Yang, 2000). These tasks can take either linear or hierarchical approaches, with the latter taking into account structural representation of topics within documents (Glavaš and Swapna, 2020).

Generally, the development of neural models from scratch for text segmentation tasks requires large training datasets (Koshorek et al., 2018) and high computational costs. In response, researchers have turned to pre-trained deep transformer models such as BERT, which offers high performance on NLP tasks and the possibility of fine-tuning its base model towards specific domains. Various transformer-based model architectures and linear, hierarchical, and multilevel models have been explored and evaluated for their performance on text segmentation.

For domain-independent models, Lukasik et al. (2020) introduced three new BERT architectures to segment documents and discourses by predicting on break points instead of classifying every piece of text. These novel architectures showed that a simple cross-segment BERT model using only local context (sequences of tokens before and after a potential break point) can perform as competitively as more complex hierarchical BERT models. Yoong et al. (2021) also developed three BERT models—BERT-NSP, BERT-SEP and BERT-SEGMENT—to perform a text tiling task (dividing a document or dialogue into semantically coherent text segments) and demonstrated that BERT-SEP, which considers the relatedness of adjacent sentences as well as information from the whole document, outperformed graph-based or bi-directional LSTM (Long Short-Term Memory) models. Lo et al. (2021) developed a two-level transformer framework incorporating language-specific or domain-specific pre-trained BERT transformers as sentence encoders, which outperformed state-of-the-art text segmentation models on a semantic coherence measure.

To develop domain-specific models, often with limited labeled training data, researchers have tested how transformer-based language models pre-trained on large amounts of general-domain data can be leveraged and adapted for a specific domain. To extract content elements from regulatory filings and property lease agreements, Zhang et al. (2020) segmented documents into paragraphs and trained BERT at the paragraph level, which achieved reasonable accuracy. They also found that training with fewer than 100 documents was sufficient to achieve an F1 score similar to that of the same model trained with the entire set of documents. Araci (2019) introduced FinBERT, a fine-tuned BERT model for the financial domain, by conducting additional pre-training and fine-tuning of BERT using text from financial news articles. FinBERT outperformed other pre-trained models with as few as 250 training examples in a sentiment analysis task involving financial phrases.

We go beyond the works mentioned that only provided information retrieval, topic segmentation, or document summarization to extract any targeted section that a social science researcher needs through a quick and manual-labor saving framework. Building on the above related works, we focus on refining a generic transformer model

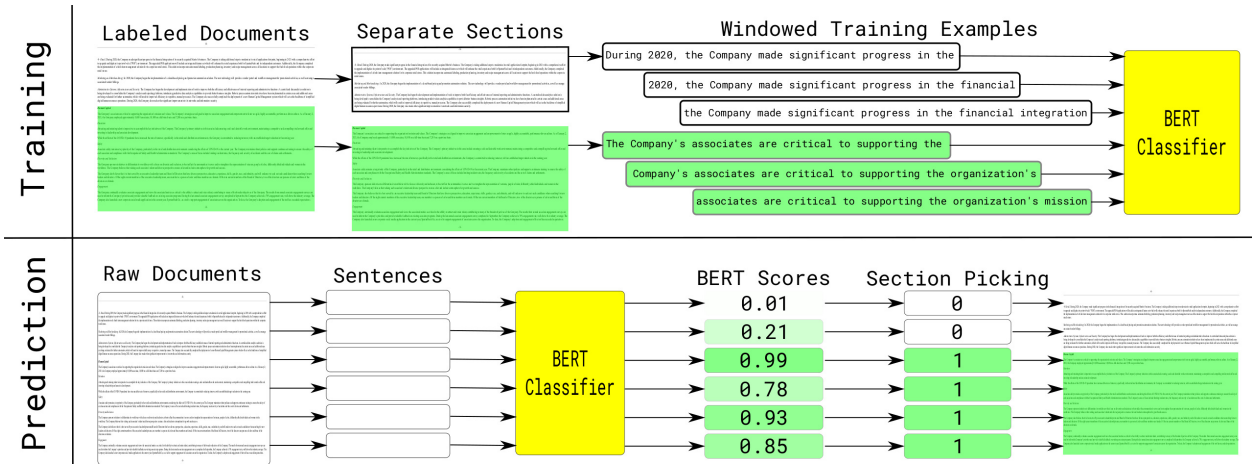

Figure 1: An illustration of the Assisted Neural Text Segmentation (ANTS) framework. In the Training stage (top pane), we fine-tune a pre-trained BERT model for text classification on hand-labeled documents that are separated into pertinent (green) and non-pertinent (white) sections. We augment our collection of training examples by creating sliding windows of tokens from the labeled sections. In the Prediction stage (bottom pane), we use our fine-tuned model on unseen documents to compute classification scores on individual sentences. We utilize these scores with a threshold method to identify a single, continuous section of relevant text for each document.

on a single domain-specific task to extract a targeted section of text in a low-resource setting. We use this test case to illustrate a framework that a social science researcher can use in place of training and recruiting manual labor.

3 Data

The Securities and Exchange Commission (SEC) requires that publicly traded companies file financial disclosures at regular intervals. The Form 10-K is an annual report that contains information about a company's business operations, financial results, and management. In November 2020, the SEC began to require its registrants to include a disclosure of their human capital resources in their Form 10-K. This resulting section is of interest to accounting researchers who want to characterize how firms discuss their human capital and whether specific diversity metrics are divulged (Choi et al., 2022).

We use MTurk to train workers to identify the described Human Capital Disclosures (HCD) section in 393 Form 10-K documents filed by S&P 500 firms from November 2020 through March 2021. The HCD section is a single, continuous segment of text located within each Form 10-K. It appears under various titles (e.g., "Human Capital", "Human Resources", "Talent", "Employee Engagement"), which span a range of sub-section topics (e.g., hiring, benefits and compensation, diversity, culture) of different lengths and combinations. We employ human labor for this extraction task due to this lack of uniformity in the section names, content, and location of the HCD section among Form 10-K documents. We use this manually collected data as a test case for our ANTS framework. In later sections, we will describe how documents are randomly sampled to obtain training and test sets to evaluate our framework.

4 Methods

In this section, we describe the ANTS framework (outlined in Figure 1) in more detail and how it is used in the scope of our specific test case. Our framework expands upon the general structure and methodologies of machine learning systems. We employ a few strategies within the framework to maximize the performance of our fine-tuned model on our task without adding more labeled documents. In training, we explore a windowing method to expand the size of our input data. In prediction, we use an approach combining the prediction scores of individual sentences and blocks of sentences to optimize our ability to locate the targeted single, continuous HCD section. Our implementation of the methods described below can be found at darc.stanford.edu/ants.

4.1 Label Training Data

To begin, documents are manually annotated to be used as inputs for training (green boxes in Training panel of Figure 1). We discuss our training inputs as documents to match how this framework

might be used by researchers who are more familiar with handling and labeling whole documents. For our specific problem, MTurk workers manually identified a single HCD section within Form 10-K documents.

After manual labeling, we separate the text within our documents into positive and negative sections. In this case, the positive section is the HCD section and the negative section is the rest of the document. Since a given HCD section may be relatively scarce in content (∼1000 tokens on average from the collected sample), we increase the amount of training examples in our dataset by windowing over each section. In this approach, illustrated in the Training panel of Figure 1, given a specific window size N, we take the first N tokens of the positive section (in green), and label that window as 1. This is the first training example for our model. Next, we move one token over, and take another window of N tokens. This is repeated until there are no more tokens in the section. We perform the same windowing for the negative section (in white), except with a label of 0. In our test case, we use a window size of 34 tokens to coincide with the median number of tokens per sentence in our sample of documents. The window size hyperparameter can be varied depending on the use case, where smaller windows might contribute too little context for the model to learn on, while larger windows might provide too few examples.

4.2 Fine-tune BERT

To fine-tune BERT, we use the implementation of BERT for binary classification from Wolf et al. (2020) We train only the final classification layer with a batch size of 32 and a learning rate of $1e^{-5}$ for 4 epochs on a 3:1 (negative:positive) balanced training set, selecting the best model based on the validation set performance. All other layer weights in the model are frozen. The training dataset was split 9:1 for training and validation. After the training/validation split, we balance the training set with a 3:1 ratio of negative to positive examples. This balancing is accomplished by under-sampling negative examples to achieve the desired ratio.

We use GPU resources on Google Colab for the initial exploration and development of our training framework, and a high performance computing cluster for final training. We run the final training using a single GPU on the Stanford High Performance Computing Sherlock cluster.

To better represent the range of performance of the fine-tuned models from our training framework in this paper, we take a random sample of input documents from the available set of labeled documents and fine-tune a BERT model using that sample instead of the full set of labeled documents. We denote a model trained on a random sample of input documents by Model$_i$, for $i = 1, \ldots, M$, where M is set to 20 in our examples. For Model$_i$, we randomly sample a number of training documents, and use the remaining documents as the test set for that model. In the prediction phase, we pick the epoch with the least validation loss during training for every Model$_i$.

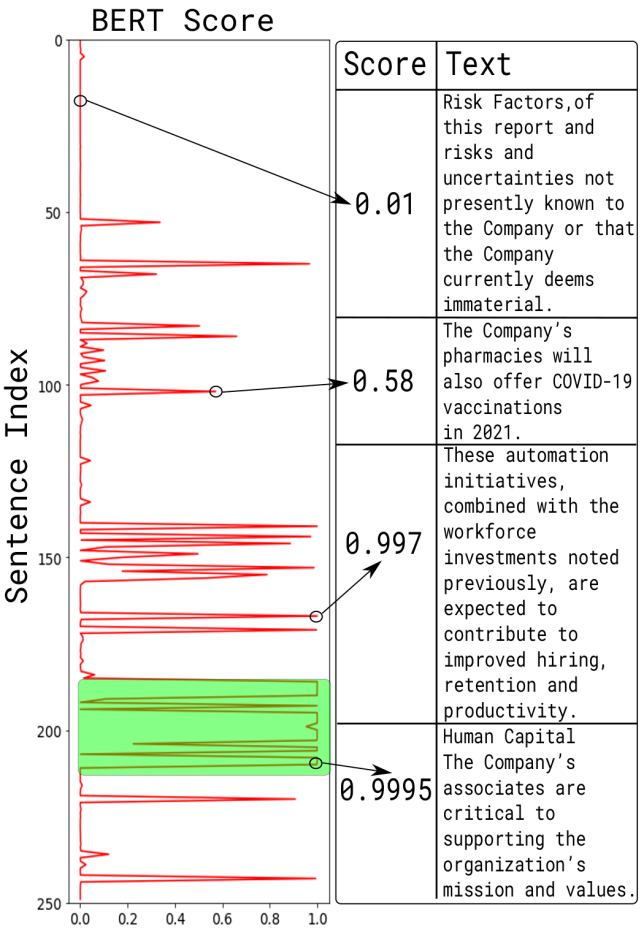

Figure 2: Example of single sentence (1-block) probabilistic scores (red line) generated during the prediction phase for a single document accompanied by sample sentence text. High scores (such as 0.9995) indicate a strong correlation with the language of the positive training examples and low scores (such as 0.01) indicate the opposite. The targeted Human Capital Disclosure (HCD) section is highlighted in green.

4.3 Classify New Text

Now that we have a fine-tuned BERT model to identify the language we are looking for, we use that model to classify text in unseen documents. For each document, we tokenize the text into sentences (Prediction panel of Figure 1) using the Python NLTK package (Bird et al., 2009) in preparation for prediction on the sentence-level. We choose to predict on sentences as they represent the most natural breaking points within a document. As an output, the model provides a probabilistic classification score for each sentence.

In our test case, we generate prediction scores for both single sentences (1-block) and for larger blocks of five sentences (5-block). In the 5-block instance, we create blocks of 5 sentences with a step size of 1 sentence, similar to the windowing strategy in training. Our rationale for generating scores for 5-blocks is to take advantage of the additional context provided by neighboring sentences to help classify the central sentence. This way, shorter sentences with less contextual information, but that are still part of the target section, are more likely to be classified properly in the method we use to identify HCD sections from the model output. This strategy is another demonstration of maximizing the available information from a scarce amount of data.

4.4 Identify Targeted Section

After running prediction and obtaining scores for each sentence in a document, we need to make a decision on which sentences are associated with our section of interest. In some use cases, where a straightforward categorization of individual sentences is sufficient, a simple threshold can be chosen to make this decision. This threshold can be tuned based on the desired outcome metrics. In our test case, where we need to find a single, continuous section of text, a more complex approach is necessary.

The choice of a section identification method is complicated by the distribution of sentence scores generated by the model. Figure 2 illustrates the score distribution (in red) as a function of sentence index for an example document. The targeted HCD section is highlighted in green. Generally, higher scores indicate a strong correlation with the language of the positive training examples and lower scores indicate the opposite. There are a few situations to consider.

First, there are sentences that are part of the targeted section and should become true positive predictions. However, the distribution of scores within the highlighted green area shows that there are individual sentences with lower scores that could end up as false negative predictions. These may simply be shorter sentences with less contextual information or they could actually contain irrelevant text, but happen to be in the targeted section. In our task of capturing the HCD section as presented in each Form 10-K document, we want to capture these sentences.

Furthermore, there are sentences with relatively high scores (such as the 0.997 example sentence and many of the other peaks outside of the highlighted green area) that contain relevant content based on the provided training examples, but are not contained within the targeted section. This is not unexpected in our case as companies are required to discuss their human capital resources in Item 1 of their Form 10-K, but this does not forbid them from discussing related content outside of Item 1. This situation can lead to many false positive predictions in our case, that could actually be relevant data in a different use case.

Finally, there are sentences with scores in the middle (such as the 0.58 example sentence). These could appear within or outside of the targeted section and the chosen method must accommodate these sentences.

In consideration of these factors, we use the combined information provided by the 1-block and 5-block scores to determine the predicted single, continuous HCD section for each test document. To start, we calculate a threshold for which to evaluate the output scores by compiling the 1-block scores produced by the Model_i given a particular set of parameters and taking the median value of scores that are less than or equal to 0.5. For each document, we then find the longest continuous section of 1-block scores that fall under that threshold. After that, we seek the longest continuous section of 5-block scores that fall under the threshold and has an overlap with the longest 1-block section. The sentence endpoints of this 5-block section determine the predicted HCD section for each document. We believe this approach provides the best balance in our attempt to capture as much of the true HCD section as possible.

5 Results

Our results are reported using the aforementioned sample of 393 Form 10-K documents from S&P 500 companies. We use three evaluation metrics: precision, recall, and Jaccard index. Precision represents how well our section identification algorithm captures positive sentences, penalizing the situation where sentences outside of the true HCD section are determined to be part of that section. In practice, however, extraneous sentences at the outside edges of the HCD section may be acceptable if most of the section itself is correctly labeled. For this, we rely on the recall score, which represents how much of the targeted section is captured. Finally, to capture both the precision and recall metrics together, we use the Jaccard index, which penalizes both false positives and false negatives.

We calculate the three metrics described above for each predicted HCD section from a document varying the number of training documents for each Model$_i$. To characterize the performance of Model$_i$, we take the mean score from all predicted documents. Documents without a predicted HCD section receive a score of zero for each metric. The plots in Figure 3 show these mean scores.

5.1 Training on Sentences versus Windows

To illustrate the effectiveness of the windowing method described earlier in the labeling phase of training, we test the ANTS framework by constructing training examples using windowing and no windowing (sentence-only) training datasets. For the no windowing model, we tokenize the separated positive and negative sections into sentences, each of which then constitutes a single training example for the BERT model. The training datasets for Model$_i$ are created from the same set of documents and the evaluation metrics are derived from predictions on the remaining documents not used in training. For sentence-level and window-level approaches, the training and test sets used in training and evaluating Model$_i$ are the same. All other hyperparameters are held constant.

Figures 3a (sentence training) and 3b (window training) show the three chosen evaluation metrics as a function of the number of training documents ranging from 1 to 19 documents with a step size of 2 and using just the 1-block scores to predict the HCD section. In other words, we choose the longest continuous section of 1-block scores that fall under the threshold described earlier. For this particular comparison, we omit the usage of 5-block scores to focus on the difference achieved just with windowing. Each dot in the displayed score distribution represents the performance of Model$_i$ for each number of training documents and the dashed lines represent the trend of the mean score value of all 20 Model$_i$'s.

A few notable differences can be seen between Figures 3a and 3b. First, we see a sharp contrast in the distribution of scores across Model$_i$'s at any given document size. In the no windowing case, a model's Jaccard index for a given Model$_i$ can range anywhere from zero to around 0.7. Strikingly, this wide spread can be observed anywhere from a number of training documents of 7 documents all the way to 19 documents. Although the overall mean performance (dashed line) displays an upward trend, this spread illustrates that if the "wrong" N documents are chosen for sentence-only training, then poor results may be observed even with large N. The score distributions in the windowed case are much narrower, mitigating the impact of selecting any particular documents for training.

Additionally, though less dramatic than the spread, there is an overall improvement in performance across the three metrics in the windowed models versus the sentence-only ones. In particular, the performance of the models trained with windows saturates after only a training input size of about 5 documents or so. The same cannot be said, and is also difficult to observe, in the models trained with sentences. This is likely caused by the substantial increase in effective training data resulting from the windowing method, which leads to a lower requirement on the number of documents needed for training.

Based on the observations above, using the windowing method during training in the ANTS framework is an effective way of improving the predictability and overall performance of the resulting fine-tuned model. At the same time, it reduces the number of manually labeled training documents required.

5.2 Varied Number of Training Documents

We train our model on various training input sizes, measured by the number of documents used. As mentioned earlier, we choose document as the input size unit to match how this framework might be used by social science (particularly business)

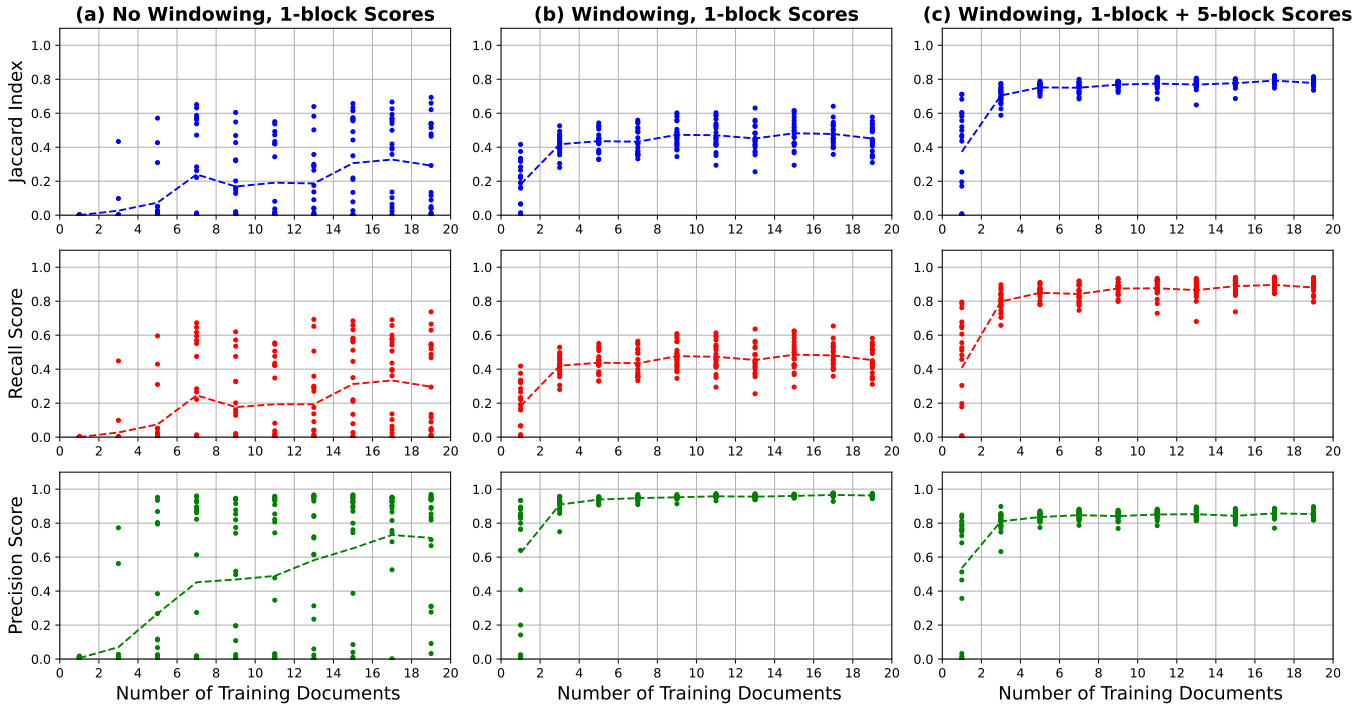

Figure 3: The x-axis represents the number of training documents and each dot in the graph represents a mean score for all test documents for Model$_i$ trained on a random subsample of the input labeled documents. The mean Jaccard (blue), recall (red), and precision (green) scores of predicted Human Capital Disclosure (HCD) sections are calculated across a set of test documents. Test set for Model$_i$ consists of the remaining documents not used for training of Model$_i$. The dashed lines show the trend of the mean score value for all Model$_i$ through Model$_M$, where $M = 20$ at each training input size. (a) Results for training on sentences only and using just 1-block scores to predict the HCD section. (b) Results for training on windows and using just 1-block scores to predict the HCD section. (c) Results for training on windows and using both 5-block and 1-block scores to predict the HCD section.

researchers who are more familiar with handling collections of documents. We use Figure 3c to discuss our findings. Figure 3c shows the same metrics described for Figures 3a and 3b, but using windowing during training and the aid of 5-block scores in finding the right section. The effectiveness of employing 5-block scores is described in the next section.

Based on the score distributions shown, there is a noticeable increase in performance as a function of the number of training documents used for fewer than 5 documents, but then a clear saturation after that point. This same behavior was noted for the results in Figure 3b, but is more evident here. Furthermore, the spread in the scores across Model$_i$'s also reduces markedly as a function of the number of training documents, also steadying at around 5 documents. This indicates that after a certain number of training documents, the predictability of the model performance is quite stable, which provides some leeway as to which training documents are chosen.

Of particular interest from a practical perspective is that the point of saturation for both performance and spread is reached at only around 5 documents. There is merely a 3.4% difference in mean Jaccard index between training on 5 documents versus 19 documents. This somewhat unexpected result illustrates that for a defined section identification task like this one, not very much training data is necessary in the ANTS framework for a fine-tuned BERT model to achieve adequate classification performance. Moreover, at 5 training documents, the model already captures part of the targeted section in 96% of the unseen documents.

5.3 Using 5-block Scores in Section Identification

As discussed in Methods, we use a combination of 1-block and 5-block scores to optimize the prediction of the HCD section for each document. The differences in Figures 3b and 3c emphasize the validity of this approach. To start, there is a clear gap in performance as reflected by the mean Jaccard

index after the point of saturation (5 documents) is reached. In the case of using only 1-block scores, the Jaccard index ranges from 0.43 to 0.48. However, in the case of using both 1-block and 5-block data, the Jaccard index ranges from 0.75 to 0.79. Looking closer at the precision and recall scores, it is clear that this dramatic difference in Jaccard index can be attributed to the sizable improvement in recall shown in Figure 3c. In fact, the precision scores are higher in the case of using only 1-block scores in Figure 3b. This means that the chosen section identification algorithm captures more of the targeted section during prediction, but at the expense of falsely including sentences just outside the edges of the section. For our test case, this is acceptable and for other use cases, this can be tuned.

As a result of calculating 5-block scores for this section identification method, the overall time spent during the prediction phase is longer. However, the performance gains for our use case are significant and additional annotated data is not required. The results here further illustrate the possibility and effectiveness of stretching out scarce text data in this framework.

5.4 Utility of "False Positives"

The section identification scheme that we choose de-emphasizes other sentences that have high classification scores, but lay outside of the actual HCD section. For instance, the sentence with score 0.997 outside of the highlighted green area in Figure 2. However, these resulting "false positive" sentences could be relevant content to a researcher even though they do not fall in the targeted section. We perform a text similarity analysis to determine whether these sentences are indeed relevant. To do this, we divide the sentences of each document into 3 categories:

1. **Actual Positive** sentences identified by workers to be part of the HCD section,

2. **False Positive** sentences determined by the 1-block model to be positive, but did not fall into the HCD section, and

3. **Negative** sentences determined by the 1-block model not to be positive and did not fall into the HCD section.

We remove English stop words from the text and then compute a TFIDF matrix for each of the three categories using the Python Scikit-learn package (Pedregosa et al., 2011). We then calculate the cosine similarity between the matrices. Notably, we find that the similarity between Actual Positive and False Positive text is very high (0.88) relative to the same measure between Actual Positive and Negative (0.38) text. For False Positive and Negative text, the similarity is 0.42. This supports the idea that the model potentially captures text relevant to the HCD section that is ignored in the scope of this paper, but may still be of value in a different context.

6 Conclusions

In this paper, we propose a practical framework to extract continuous segments of text from unstructured documents, with a particular focus on text-intensive research in business and social science. The ANTS framework utilizes a pre-trained BERT model to identify targeted sections 96% of the time using only 5 training examples. This general framework can enable subject matter experts to accelerate their research by reducing the time commitment needed to extract large amounts of relevant text given a very small number of training examples. Our proof of concept using Human Capital Disclosure sections of SEC filings demonstrates that manually coding only a few documents provides enough training data for a model to effectively identify the relevant section of the remaining documents. Furthermore, the success of this framework opens a number of other valuable research questions from the same documents. For instance, what distinguishes the official HCD text from thematically similar data (as flagged by ANTS) in the remainder of the document? Or, how did companies report human capital information prior to the SEC's disclosure requirement?

The ANTS framework provides the opportunity for researchers to use additional domain knowledge to integrate the sentence-level scores from a trained model. In this report we use a section picking algorithm that is constrained to identify only a single contiguous section to mirror the SEC filing structure. An ANTS framework that could, for instance, identify boilerplate language from corporate charters, could be tuned based on the known length, location and number of boilerplate sections. The wealth of trained models also provides the opportunity to extract or flag relevant text from semi-structured documents (e.g., HTML), spoken

text transcriptions or social media posts.

While our proof of concept only returns flagged sections as an output, it suggests an application as a successor to "word list" based research methods. Work such as Loughran and Mcdonald (2011) describes counting words and phrases as a proxy for an underlying theme, such as uncertainty. Using ANTS, researchers can identify sections of interest and prepare document scores from aggregating model results. Taking human capital disclosures and diversity as an example, ANTS could be trained with language on workforce diversity from SEC filings, and then used to generate a diversity score by counting the number of sentences discussing diversity. This work could circumvent the technical and arduous task of building word lists, and provide context aware metrics that can flag a diversifying workforce without false positives from a diversifying supply chain.

Taken together, the ANTS framework demonstrates a rich set of avenues that can be used to accelerate, augment and amplify the work of academic researchers in the social sciences. As deep learning tools are released on free and reduced cost platforms (e.g., Colab, OpenAI, HuggingFace), researchers will build effective datasets from larger, more diverse and more subtle text sources. We hope that ANTS can be leveraged to facilitate this growth in text data and democratize deep learning advances in new and unexpected ways.

7 Acknowledgment

Some of the computing for this project was performed on the Sherlock cluster. We would like to thank Stanford University and the Stanford Research Computing Center for providing computational resources and support that contributed to these research results.

References

Dogu Araci. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. *ArXiv*, abs/1908.10063, 8 2019. URL https://arxiv.org/abs/1908.10063.

Sebastian Arnold, Rudolf Schneider, Philippe Cudré-Mauroux, Felix A Gers, and Alexander Löser. SECTOR: A Neural Model for Coherent Topic Segmentation and Classification. *Transactions of the Association for Computational Linguistics*, 7:169–184, 2019. doi: 10.1162/tacl/a/00261. URL https://aclanthology.org/Q19-1011.

Dennis Aumiller, Satya Almasian, S Lackner, and Michael Gertz. Structural text segmentation of legal documents. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, 2021. URL https://doi.org/10.1145/3462757.3466085.

Steven Bird, Edward Loper, and Ewan Klein. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.

David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003. URL https://dl.acm.org/doi/10.5555/944919.944937.

Jung Ho Choi, Joseph Pacelli, Kristina M. Rennekamp, and Sorabh Tomar. Do Jobseekers Value Diversity Information? Evidence from a Field Experiment. *SSRN Electronic Journal*, 2022. ISSN 1556-5068. doi: 10.2139/ssrn.4025383.

Wesley T Chuang and Jihoon Yang. Extracting sentence segments for text summarization: a machine learning approach. *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 152–159, 2000. URL https://doi.org/10.1145/345508.345566.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv*, abs/1810.04805, 2019. URL https://arxiv.org/abs/1810.04805.

Goran Glavaš and Somasundaran Swapna. Two-level transformer and auxiliary coherence modeling for improved text segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/6284.

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. Text Segmentation as a Supervised Learning Task. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2:469–473, 3 2018. doi: 10.18653/v1/n18-2075. URL https://arxiv.org/abs/1803.09337v1.

Kelvin Lo, Yuan Jin, Weicong Tan, Ming Liu, Lan Du, and Wray Buntine. Transformer over Pre-trained Transformer for Neural Text Segmentation with Enhanced Topic Coherence. *ArXiv*, abs/2110.07160, 2021. URL https://arxiv.org/abs/2110.07160.

Tim Loughran and Bill Mcdonald. When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. *Journal of Finance*, 66(1):35–65, 2 2011. ISSN 00221082. doi: 10.1111/J.1540-6261.2010.01625.X.

Michal Lukasik, Boris Dadachev, Gonçalo Simões, and Kishore Papineni. Text Segmentation by Cross Segment Attention, 2020. URL https://arxiv.org/abs/2004.14535.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 1 2013. doi: 10.48550/arxiv.1301.3781. URL https://arxiv.org/abs/1301.3781v3.

Minh Tien Nguyen, Dung Tien Le, and Linh Le. Transformers-based information extraction with limited data for domain-specific business documents. *Engineering Applications of Artificial Intelligence*, 97:104100, 1 2021. ISSN 0952-1976. doi: 10.1016/J.ENGAPPAI.2020.104100. URL https://www.sciencedirect.com/science/article/pii/S0952197620303481.

Hyo Jung Oh, Sung Hyon Myaeng, and Myung Gil Jang. Semantic passage segmentation based on sentence topics for question answering. *Information Sciences*, 177(18):3696–3717, 9 2007. ISSN 0020-0255. doi: 10.1016/J.INS.2007.02.038. URL https://doi.org/10.1016/j.ins.2007.02.038.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Yla R. Tausczik and James W. Pennebaker. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of Language and Social Psychology*, 29(1):24–54, 12 2009. ISSN 0261927X. doi: 10.1177/0261927X09351676. URL https://journals.sagepub.com/doi/abs/10.1177/0261927x09351676.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, 10 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Siang Yun Yoong, Yao Chung Fan, and Fang Yie Leu. On Text Tiling for Documents: A Neural-Network Approach. *Lecture Notes in Networks and Systems*, 159 LNNS:265–274, 2021. ISSN 23673389. doi: 10.1007/978-3-030-61108-8/26. URL https://link.springer.com/chapter/10.1007/978-3-030-61108-8_26.

Ruixue Zhang, Wei Yang, Luyun Lin, Zhengkai Tu, Yuqing Xie, Zihang Fu, Yuhao Xie, Luchen Tan, Kun Xiong, and Jimmy Lin. Rapid Adaptation of BERT for Information Extraction on Domain-Specific Business Documents. *ArXiv*, abs/2002.01861, 2 2020. ISSN 2331-8422. URL https://arxiv.org/abs/2002.01861v1.