2018-01-01

Suggesting Missing Information in Text Documents

Grant Michael Hodgson
Brigham Young University

Suggesting Missing Information in Text Documents

Grant Michael Hodgson

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Kevin Seppi, Chair
Christophe Giraud-Carrier
Seth Holladay

Department of Computer Science

Brigham Young University

ABSTRACT

Suggesting Missing Information in Text Documents

Grant Michael Hodgson
Department of Computer Science, BYU
Master of Science

A key part of contract drafting involves thinking of issues that have not been addressed and adding language that will address the missing issues. To assist attorneys with this task, we present a pipeline approach for identifying missing information within a contract section. The pipeline takes a contract section as input and includes 1) identifying sections that are similar to the input section from a corpus of contract sections; and 2) identifying and suggesting information from the similar sections that are missing from the input section. By taking advantage of sentence embedding and principal component analysis, this approach suggests sentences that are helpful for finishing a contract. We show that sentence suggestions are more useful than the state of the art topic suggestion algorithm by synthetic experiments and a user study.

Keywords: natural language processing, machine learning, contracts, missing information

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the late 1960s Aluminum Company of America (ALCOA) entered into a long-term agreement to provide aluminum for Essex Group, Inc. (Essex) [1]. As part of the agreement, they included a complicated pricing formula that was meant to fluctuate proportionally with the price of aluminum production. The pricing formula worked well for a number of years. In the 1970s, however, the Organization of the Petroleum Exporting Countries (OPEC) took actions to increase oil prices. The increase in oil prices greatly increased the cost of electricity and therefore the non-labor costs of ALCOA's aluminum production. The pricing formula used in ALCOA's agreement with Essex failed to account for this price increase, and ALCOA was faced with potential losses of $75 million. To avoid the loss, ALCOA was forced to enter into costly litigation over the contract [1]. If the contract drafters had thought to include fluctuations in the price of electricity, perhaps litigation over the agreement could have been avoided.

A key part of contract drafting involves thinking of issues that may arise and defining the obligations of each party should the issue arise over the term of the contract. In other words, attorneys drafting a contract often spend time trying to identify what might be missing from a contract. Costly litigation may be more easily avoided if it were easier to identify and suggest missing information from contracts.

Compared to other types of documents, legal documents and specifically contracts have unique properties. Similar types of contracts tend to address similar types of issues. Similar types of contracts also tend to share structure and address similar issues in the same sections as other contracts. This formulaic characteristic of contracts can be leveraged to identify what might be missing from a contract. Comparing one section in a contract (an input section) to many other similar sections from other contracts may allow a user to identify what may be missing from the input section.

Comparing documents to determine what is different may be useful for other types of legal documents as well. Although this paper focuses on contracts, we believe that our solution to identifying what text is missing from a document would be useful for other legal documents or other types of documents that follow a well-defined structure. It may also be of interest to people who are unable to pay for expertise to draft a legal document. Our solution could help them identify what may be missing in the document that they are using.

Previous approaches to suggesting missing information in a text document involve suggesting topics. Topics are usually one or two words and thus cannot convey detailed information. Even if a topic is suggested as missing the author may not necessarily know what to say about the topic. In contrast, we will show that suggesting longer sequences of words (such as sentences) can be much more useful.

We present a pipeline approach for identifying missing information within a document by comparing the document to other similar documents. The pipeline will take a document as input and will include 1) identifying documents that are similar to the input document; and 2) identifying and suggesting sentences from the documents that are missing from the input document. We present a technique that suggests missing information to a user who can then choose whether to include the information in the document. In chapter 2 we will discuss previous work that is related to suggesting missing information. In chapter 3 we will discuss our approach in more detail. In chapter 4 we will discuss the results of our approach. Chapter 5 contains our conclusion and potential future work.

Chapter 2

Related Work

Identifying what is missing in a text document (e.g. a contract) is related to several research areas in computer science such as inpainting, contract mining, document clustering, text summarization, sentence embeddings, automatic topic suggestion, and translation using deep neural networks. We briefly discuss each of these topics.

2.1 Inpainting and Scene Completion

Image processing has some similarities with finding missing information. Inpainting involves restoring missing portions of an image to make it more visually plausible [7]. Typical applications include removing logos from videos, digital reconstruction of images that have faded, removing an object or person from an image, and filling in the space leftover [4]. Inpainting can be used to fix scratches, stains, or other large-scale missing regions by interpolating based on other portions of the image that are not missing [4].

Scene completion is a related task that replaces a portion of an image with another visually plausible portion from a different image [6]. A scene completion algorithm may use semantic scene matching and local context matching to fill in missing objects in a picture [6]. Specifically, it involves finding a "subset of images depicting semantically similar scenes," finding "patches in [the] subset that match the context surrounding the missing region," and blending "in the most similar patches" [6]. To succeed at scene completion, "context encoders need to both understand the content of the entire image, as well as produce a plausible hypothesis for the missing part(s)" [14].

2.2 Contract Mining and Document Clustering

Gao et al [5]. discuss techniques for extracting information from contracts. They suggest creating a checklist based on the extracted information that can be used to make sure an attorney has accounted for every important issue. The method involves extracting the commonly occurring exception phrases for a domain of interest to build a vocabulary of exceptions that arise in each domain [5]. The extracted vocabulary could then be used as a checklist of items for a contract drafter to look for. However, the specifics of how the above-mentioned vocabulary could be used is not discussed and it does not appear to have been implemented.

In addition, our proposed technique relies partially on document clustering. Document clustering will enable us to identify documents that are similar to an input document. Using K means with term frequency inverse document frequency (TF-IDF) is a well-documented technique that has proven effective in clustering documents [15].

2.3 Text Summarization

At first glance, text summarization appears to be somewhat related to identifying missing information. Nenkova et al. [13] provide an overview of various text summarization techniques. According to their overview, nearly all summarizers do some form of the following: 1) they create an intermediate representation of the input text, capturing the key aspects of the text; 2) they score sentences of the text based on the created representation; and 3) they select several sentences based on the scores to create the final summary [13]. In some approaches, "the optimal collection of sentences is selected subject to constraints that try to maximize overall importance, minimize redundancy, and. . . maximize coherence" [13]. Text summarization has some things in common with our problem. However, it fails to solve the problem because text summarization merely creates an overall summary of all the documents without identifying what may be missing from a document.

2.4 Sentence Embeddings

Sentence embedding techniques are used to capture the semantic meaning of a sentence in a vector representation. With good vectors, sentences with similar meaning will have similar vector representations. This is useful because it allows a computer to determine semantic similarity between sentences that may be worded differently but mean essentially the same thing [9]. Several techniques exist for creating vector representations of sentences. Kiros et al [9]. use an encoder-decoder model that tries to reconstruct the surrounding sentences of an encoded passage. Their method relies on the sequential nature of sentences within a document and the theory that sentences that appear in a similar context will have similar meaning. In this way, "[s]entences that share semantic and syntactic properties are thus mapped to similar vector representations."

Kenter et al. [8] create sentence embeddings by averaging the word vectors within a sentence. However, instead of using word embeddings created by techniques such as word2vec [12] and GLoVe [16] , they optimize the training of the word embeddings for sentence representation using a Siamese continuous bag of words (CBOW) neural network. Sentence embeddings are helpful in grouping similar sentences together, allowing us to identify what kinds of sentences may be missing from a contract.

2.5 Automatic Topic Suggestion

In addition, some research has been completed on suggesting missing topics from a document. West et al. [20] propose a technique that begins with identifying topics in an input document. They then identify missing topics by generalizing from a large background corpus using principal component analysis. Finally, they rank and suggest missing topics to a user who can then decide whether to discuss them in the input document. The approach of West et al. relies heavily on Wikipedia. They use a fixed group of topics consisting of the set of all Wikipedia articles because they assume that Wikipedia's coverage is so vast that nearly any conceivable topic has a corresponding Wikipedia article. While this assumption may hold true in general, in some specific applications there may not be a Wikipedia article about every topic. For example, the merger and acquisition (M&A) domain has topics such as antitrust reverse termination fee, and no pending litigation, which as of this writing, do not contain articles in Wikipedia. In addition, it would be beneficial, especially for certain types of legal documents, to suggest sentences instead of just topics. Compared to receiving a topic suggestion, a user that receives a suggestion in the form of a sentence may require less time to integrate the suggestion into a document. We take the approach of West et al. one step further by suggesting sentences that could be added (possibly with some modification by the user) to an input document and by eliminating the need to rely on Wikipedia.

2.6 Translation Using Deep Neural Networks

Translation techniques using sequence to sequence deep neural networks also appear to be tangentially related. The problem of identifying missing text from a document can be thought of as a sequence translation problem. That is, translating from the input section to the missing text. Although this approach has never been applied to this problem, given the recent success of Deep Neural Networks (DNNs) in translation, (see Bahdanau et al. [2] and Luong et al. [11]) that approach would seem like an alternative worth considering. We use it as a comparison with our approach.

Chapter 3

Methods

Our approach for suggesting missing information includes gathering and preprocessing contract data, clustering documents (contract sections) together, and identifying important sentences that are missing from an input section by comparing the input section with a cluster of similar sections. We have chosen to suggest sentences to a user because contract language can often be copied and pasted into a new contract with only minor changes. Suggested sentences may also capture important nuances that a suggested topic might miss.

We present two new methods for suggesting missing information. We will refer to the method of West et al. simply as West and use it as the baseline for comparing the two new methods. We will also present results from a neural translation model (NTM) as applied to our problem. We use NTM as an additional comparison for completeness. We call the first new method the Missing Text Determiner (MTD). It takes advantage of sentence vectors and uses them to suggest missing sentences instead of missing topics. We call the second new method Topic Recommender System (TRS) because it uses matrix factorization techniques that have been used in recommender systems to suggest missing topics. TRS identifies missing topics, but then uses the topics to find sentences that represent the missing topics. TRS uses Latent Dirichlet Allocation (LDA) as a topic model to learn topics and matrix factorization with gradient descent to make suggestions. We describe the two new methods in more detail below.

3.1 Suggesting Sentences with Principal Component Analysis (MTD)

We call our technique for suggesting missing sentences, Missing Text Determiner (MTD). It involves i) finding documents that are similar to the input section (3.1.1); ii) creating clusters of sentences (3.1.2); iii) performing principal component analysis (3.1.3); and iv) suggesting the top N sentence clusters (3.1.4). We explain each step in further detail below. In addition, we provide an example flow chart of the steps performed by MTD in Figure 3.1.

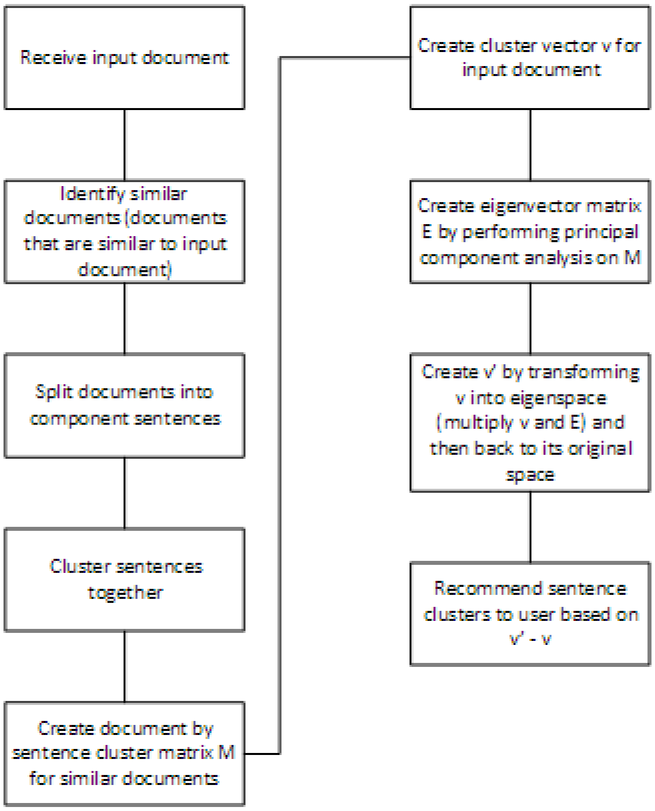

Receive input document
Identify similar documents (documents that are similar to input document)
Split documents into component sentences
Cluster sentences together
Create document by sentence cluster matrix M for similar documents
Create cluster vector v for input document
Create eigenvector matrix E by performing principal component analysis on M
Create v' by transforming v into eigenspace (multiply v and E) and then back to its original space
Recommend sentence clusters to user based on v' - v

Figure 3.1: This flowchart shows the steps performed in Missing Text Determiner (MTD)

3.1.1 Finding Similar Documents

We begin by finding the documents that are most similar to the input document. We use term frequency inverse document frequency (TF-IDF) to create a vector representation of the input document and each document in the corpus. Then we compute the cosine distance between the input document and each document in the background corpus. Our algorithm then uses only the N closest documents in the following steps (we will refer to these documents as the similar documents).

3.1.2 Sentence Clustering

We create a list containing all sentences from the similar documents using the Natural Language Tool Kit's Punkt tokenizer [3]. However, some processing is needed because sentences in contracts tend to be much longer than average sentences in English. They often contain enumerated lists like the following:

> This Plan of Merger has been duly executed and delivered by, and (assuming due authorization, execution and delivery by Purchaser) constitutes valid and binding obligations of, Company and is enforceable against Company in accordance with its terms, except to the extent that (i) such enforcement may be subject to applicable bankruptcy, insolvency, reorganization, moratorium or other similar Laws, now or hereafter in effect, relating to creditors' rights generally and (ii) equitable remedies of specific performance and injunctive and other forms of equitable relief may be subject to equitable defenses and to the discretion of the court before which any proceeding therefor may be brought.

Although technically one sentence, these enumerated lists can often span the equivalent length of many paragraphs. Using regular expressions (see appendix A), we split each item in the list to form individual sentences. Thus each enumeration marker (e.g. (i)) becomes a sentence break.

We compute sentence embeddings for each sentence using a Siamese CBOW [8] model that we trained on the entire corpus of contracts. The sentences are clustered using the sentence embeddings with K means. Cosine distance is used as the distance metric for the clustering. This results in a group of C clusters ($C = K$, the number of clusters), each cluster having a varying number of sentences contained within. We can also identify which document each sentence originated from.

3.1.3 Principal Component Analysis

Next, we create a matrix M for use in principal component analysis (PCA). The documents D make up the rows, while the clusters C make up the columns, creating a $D \times C$ matrix. For each position m_{ij} in M we count the number of sentences that document d_i has in cluster c_j. This results in a matrix where most values are 0s or 1s but occassionally a higher value will appear.

Similar to West, we create a vector representation v of the input document which is the same length and is filled in the same way as a row in M. Following the PCA algorithm, we compute a covariance matrix and subtract the mean. We compute the eigenvectors and sort them in descending order of their associated eigenvalues. This creates a matrix of eigenvectors E. We take only the top K eigenvectors (the principal components) creating a matrix E_{reduce}. Finally, we transform v into eigenspace and then back to its original space with the following equation.

$$v' = (vE_{reduce}^T)E_{reduce} \tag{3.1}$$

3.1.4 Sentence Suggestion

Similar to West, which suggests missing topics, v' will allow us to determine which types of sentences are missing from v. We can create a reconstruction gain vector (as named in West) with $v_{suggestion} = v' - v$. This provides a ranking of each sentence cluster. If the jth value v'_j in the output vector v' is high, but the corresponding value v_j is low then the cluster c_j is a suggested cluster. Thus, a user should consider adding a sentence similar to a sentence found in the cluster c_j to input document D.

3.2 Topic Recommender System

The Topic Recommender System (TRS) uses a topic modeling approach to find missing topics in an input document and then finds sentences that represent the missing topics. One deficiency of West's method is that its topics are obtained from Wikipedia article titles. Thus, topics not contained in Wikipedia cannot be suggested. To eliminate the dependency on Wikipedia, we propose using Latent Dirichlet Allocation (LDA) to generate topics within a cluster of similar documents. LDA defines a topic as a distribution over words and describes a document as a distribution over topics.

TRS first computes a LDA model using all of the documents in a corpus. It then creates a matrix R of size $D \times T$ where D is the number of documents in the corpus and T is the number of topics. We use contract sections as documents as explained above. A document will be deemed to contain a topic if the probability of the topic appearing in the document is above a predefined threshold. Thus, each entry r_{ij} in R will contain a 1 if document d_i contains topic j and 0 if it does not.

We use matrix factorization with gradient descent to identify and suggest missing topics from an input document. Matrix factorization with gradient descent is a well-known technique that has been described by Lee et al.[10] and was successfully used in a recommender system by Takacs et al.[17] We follow the approach as described by Yeung [21]. For completeness, in the remainder of this section we reproduce his implementation of matrix factorization with gradient descent. We assume that we would like to discover K latent features that determine whether a document contains any given topic. In matrix factorization, we seek to create two matrices, P (size $|D| \times K$) and Q (size $|T| \times K$), that when multiplied together, create a new matrix R' that approximates R with missing values filled in. The filled in values will indicate whether a topic should be included in a document or not. To create P and Q, we initialize them with random values and then use gradient descent to adjust their values until the difference between R' and R is minimized. The error can be computed as

$$e_{ij}^2 = (r_{ij} - r'_{ij})^2 = (r_{ij} - \sum_{i=1}^{k} p_{ik} q_{ik})^2 \tag{3.2}$$

The gradients are then obtained with the following two equations.

$$\frac{\partial}{\partial p_{ik}} e_{ij}^2 = -2(r_{ij} - r'_{ij})(q_{kj}) = -2e_{ij}q_{kj} \tag{3.3}$$

$$\frac{\partial}{\partial q_{ik}} e_{ij}^2 = -2(r_{ij} - r'_{ij})(q_{kj}) = -2e_{ij}p_{ik} \tag{3.4}$$

The values of P and Q are then updated with the following equations until convergence.

$$p'_{ik} = p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + \alpha(2e_{ij}q_{kj} - \beta p_{ik}) \tag{3.5}$$

$$q'_{ik} = q_{ik} + \alpha \frac{\partial}{\partial q_{kj}} e_{ij}^2 = q_{kj} + \alpha(2e_{ij}p_{ik} - \beta q_{kj}) \tag{3.6}$$

where β is the regularization parameter. We use scikit-learn's implementation of non-negative matrix factorization with the default parameters [15]. The resulting matrix R contains a row v' that represents the input document. The scores in v' are an approximation of what topics should be contained in the input document. Thus, the values that are high in row v' but low in row v of the original matrix R represent topics that are missing from the original input document. These are the topics that should be suggested for adding to the original document.

We use the suggested topics to find sentences. Using the Siamese CBOW model, we compute word vectors for words in each suggested topic. By averaging the first N word vectors in the topic's distribution of words we create a pseudo sentence vector that can then be used to find similar vectors in the background corpus. However, as explained below in the results section, we discovered that the sentences found in this manner did not appear to represent the topic. We found that averaging the words in a topic was not an effective way to identify sentences that reflected the words in the topic.

Chapter 4

Results

Due to the nature of the problem we are trying to solve, evaluation can be somewhat difficult. Like scene completion and identifying missing topics, identifying and suggesting missing sentences is an inherently under constrained problem, because any text that is not an exact match of what is included in the input document can be considered missing text [6]. In addition, unlike missing portions of an image which have a fixed size, documents can be made arbitrarily large. It is difficult to know at what point a text document contains all relevant information such that it cannot be improved by adding more. Despite these difficulties, we use automatic approaches and a user study for evaluation as discussed in the following sections.

4.1 Additional Technical Details

There are some additional technical details the reader should be aware of regarding the implementation used in obtaining the following results. First, We separate some punctuation from the neighboring words so that Python's split function would return whole words with no punctuation. In addition, We create vector representations of sentences using Siamese CBOW. We trained the siamese CBOW model on our contract data using words that appear four or more times in the corpus. This makes the training easier for the Siamese CBOW model created by [8]. In total, there were approximately 40,000 vocabulary words for the data sample scraped from the Electronic Data Gathering and Retrieval System (EDGAR).

EDGAR is maintained by the Security and Exchange Commission (SEC) and is a good place to find publicly available contract data.

We scraped 10 years (2006 to 2015) of S-4 forms from EDGAR and gathered the 1200 various merger and acquisition contracts which are typically found in those filings. The s-4 forms contained a mixture of different M&A contract types from different law firms. Using regular expressions, we split the contracts at the section level. Splitting the contracts into sections yielded 97,048 sections which are the documents discussed in section 3.1.

We cluster the sentences using NLTK's implementation of K means [3]. We found that using $\frac{1}{8}$ of the number of sentences for the number of clusters worked well. We also found that using between 200 to 300 similar documents worked best. The sentence vectors were 300 dimensions.

4.2 Automatic Testing: Missing Text Determiner

We created automatic tests to obtain objective results and to enable us to perform much more testing than would otherwise be possible using humans trained in contract drafting. Our goal was to artificially create documents that were missing text by deleting a sentence from the document. By deleting a sentence, we create a document that is missing text and at the same time, we know what text is missing from the document. Thus, suggestions made by MTD can be compared with the deleted sentence to determine how well MTD is performing. We used two different types of tests to determine the effectiveness of the Missing Text Determiner (MTD).

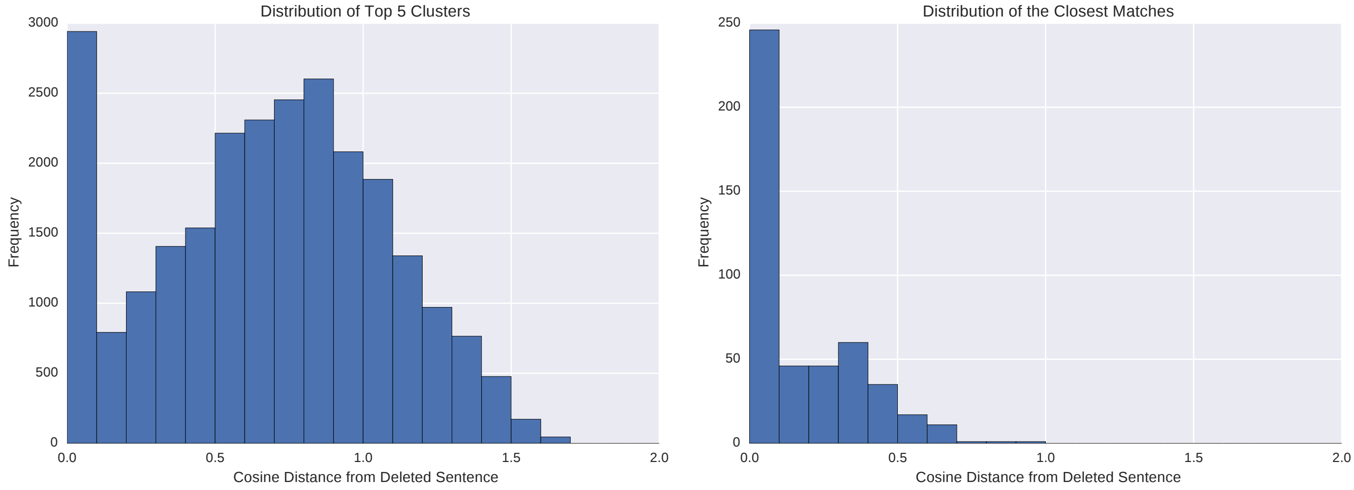
For the first test, we randomly chose an input section from the corpus and then deleted a random sentence from the section. We did not use very small sections in our testing (i.e. we did not use sections that contained 2 sentences or less). We included the original input section, which contained the missing sentence, in the group of similar documents. Thus an exact copy of the deleted sentence was always present in the documents the algorithm used. We then determined whether the algorithm suggested a sentence that exactly matched the deleted sentence within the top 10 sentence clusters that it suggests. Using 200 similar documents, we ran the test 1049 times and found an exact match 814 times, achieving an accuracy of 77.6%. Even with an exact copy of the deleted sentence in the 200 similar documents, the algorithm does not obtain perfect results. We do not know exactly why this occurs but we have some ideas. On some occasions, the algorithm may be suggesting clusters containing sentences that are similar to the deleted sentence, but due to random variation, the exact match of the deleted sentence was not placed in the best cluster by our clustering algorithm. Thus, the sentence may not be suggested because of the cluster it is placed in.

The second test is the same as the first except that the original input section was not included in the group of similar documents. Thus, an exact copy of the sentence is not guaranteed to appear in the documents used by the algorithm. We created this test to determine how well the algorithm would perform when the exact missing sentence is not contained in the background corpus. To evaluate performance under this test, we show histograms in Figure 4.1 containing the cosine distances between the deleted sentence and each of the sentences contained in the top five recommended clusters. Both histograms in Figure 4.1 contain cosine distance along the x-axis with distance frequency along the y-axis. The histogram on the left shows the distribution of cosine distances between the deleted sentence and every sentence in the top 5 recommended clusters. The histogram on the right shows the distribution of distances between the deleted sentence and the closest sentence in the top 10 suggested clusters.

To gain intuition on what different cosine distances may mean, we display some sample deleted sentences along with their closest match (the sentence that had the minimum cosine distance during the run) in table 4.1.

	Deleted Sentence	Closest Sentence	Distance
1	no Buyer Entity has repudiated or waived any material provision of any such Contract; and	no Buyer Entity has repudiated or waived any material provision of any such Contract; and	0.0
2	Conditions to the Obligations of Each Party .	Section 7.01 Conditions to the Obligations of Each Party .	0.102
3	a) The terms of each outstanding compensatory option under any agreement , plan or arrangement of Clearwire (the Clearwire Stock Option Plans) to purchase shares of Clearwire Class A Common Stock (a Clearwire Stock Option) , whether or not exercisable or vested , shall be adjusted as necessary to provide that , at the Effective Time , each Clearwire Stock Option outstanding immediately before the Effective Time will be converted into an option to acquire , on the same terms and conditions as were applicable under that Clearwire Stock Option , the same number of whole shares of Class A Common Stock (rounded down to the nearest whole share) as the holder of the Clearwire Stock Option would have been entitled to receive under the Merger had the holder exercised the Clearwire Stock Option in full immediately before the Effective Time , at a price per share (rounded up to the nearest whole cent) equal to :	8 ,862 ,169 shares of Clearwire Class A Common Stock were subject to outstanding Clearwire Stock Options , 740 ,000 shares of Clearwire Class A Common Stock were subject to outstanding Clearwire restricted stock units and 5 ,445 ,317 shares of Clearwire Class A Common Stock were authorized and reserved for future issuance under the Clearwire Stock Option Plans ,	0.158
4	Until and unless each party has received a counterpart hereof signed by the other party hereto , this Agreement shall have no effect and no party shall have any right or obligation hereunder (whether by virtue of any other oral or written agreement or other communication).	Any such counterpart may be delivered by facsimile or other electronic format (including .pdf).	0.268
5	REPRESENTATIONS AND WARRANTIES OF CAPITAL PACIFIC AND THE BANK .	ARTICLE IV REPRESENTATIONS AND WARRANTIES OF BANK	0.317
6	Except as would not reasonably be expected to have , individually or in the aggregate , a RG Material Adverse Effect , RG and its Subsidiaries are (and since January 1 , 2014 have been) in compliance with the terms of all such Permits .	b) Except as would not , individually or in the aggregate , reasonably be expected to have a Parent Material Adverse Effect , Parent and each of its Subsidiaries is , and since January 1 , 2012 has been , in compliance with ,	0.345
7	Except as set forth in Section 2.22 of the Edge Schedule , Edge is not obligated , by virtue of a prepayment arrangement , make-up right under a production sales contract containing a take or pay or similar provision , production payment or any other arrangement , to deliver hydrocarbons having a value in excess of $500 ,000 attributable to the Edge Properties at some future time without then or thereafter receiving full payment therefor .	Neither ANB nor any of the ANB Subsidiaries , nor , to the Knowledge of ANB , any other party thereto , is in breach of any of its obligations under any such agreement or arrangement , except as set forth in Section 3.4(s) of the ANB Disclosure Schedule .	0.469
8	Section 2.7 Exchange Agent , Depositary and Clearance System Arrangements .	to terminate the Starwood ESPP effective immediately prior to the Closing Date .	0.579

Table 4.1: This table gives examples of sentences across a spectrum of distances. Sentences within .15 tend to be very similar. On occasion, sentences greater than .15 are very similar (e.g. row 6) and sentences that have a close cosine distance are not as similar as might be expected (e.g. rows 3, 4)



(a) Distribution of suggested sentences.

Figure 4.1: The histogram on the left shows the distribution of cosine distances between the deleted sentence and every sentence in the top 5 recommended clusters. The histogram on the right shows the distribution of distances between the deleted sentence and the closest sentence (measured by cosine distance) in the top 10 suggested clusters. The goal was to see if the algorithm could recommend a sentence similar to one that was deleted from the input section.

It is important to note that there may be more than just the deleted sentence missing from the input section. Identifying missing text is difficult because it is possible to add any amount of text to a document. Thus even though there are many sentences suggested that do not closely match the deleted sentence (as seen in the above histograms), they may, in some situations, be useful for completing the input section.

We found that sentences that were within cosine distance of .15 tended to be very similar to each other. We make the assumption that an attorney would be able to take a sentence that is within .15 cosine distance away from a missing sentence and easily modify it to fit her needs. To show that this is true we have created Table 4.2, which shows examples of sentence pairs that were about .15 cosine distance or less. Thus, during the second test we considered each iteration that found a closest matching sentence (within the top 10 recommended clusters) .15 or less to be a hit and any iteration with a closest matching sentence greater than .15 to be a miss. Given these assumptions, we compute accuracy for the second test.

	Deleted Sentence	Closest Sentence	Distance
	no Buyer Entity has repudiated or waived any material provision of any such Contract; and	no Buyer Entity has repudiated or waived any material provision of any such Contract; and	0.0
	has used or is using any corporate funds for any direct or indirect unlawful payments to any foreign or domestic governmental officials or employees ,	has used or is using any corporate funds for any direct or indirect unlawful payments to any foreign or domestic governmental officials or employees ,	0.0
	The Company shall have delivered to Parent a certificate , dated the date of the Closing , signed by a duly authorized officer of the Company , certifying as to the satisfaction of the conditions specified in Section 8.02(a) and Section 8.02(b).	The Company shall have delivered to BioSante a certificate , dated the date of the Closing , signed by a duly authorized officer of the Company , certifying as to the satisfaction of the conditions specified in Section 8.02(a) and Section 8.02(b).	0.013
	all assets reflected on the Potomac Unaudited Interim Balance Sheet; and	all assets reflected on the Acquiror Unaudited Interim Balance Sheet; and	0.0757
	Conditions to the Obligations of Each Party .	Section 7.01 Conditions to the Obligations of Each Party .	0.102
	The obligation of CFC to consummate the Merger is also subject to the fulfillment or written waiver by CFC prior to the Effective Time of each of the following conditions :	The obligation of VCB to consummate the Merger is also subject to the fulfillment , or written waiver by VCB prior to the Effective Date , of each of the following conditions :	0.108
	elect to the Board of Directors of Parent any person who is not a member of the Board of Directors of Parent as of the date hereof ;	elect to the Board of Directors of the Company any person who is not a member of the Board of Directors of the Company as of the date hereof ;	0.113
	the directors of Merger Sub immediately prior to the Effective Time shall be the directors of the Surviving Corporation and	Each of the parties hereto shall take all necessary action to cause the directors and officers of Merger Sub immediately prior to the Effective Time to be the directors and officers of the Surviving Corporation immediately following the Effective Time , until their respective successors are duly elected or appointed and qualified or their earlier death , resignation or removal in accordance with the certificate of incorporation and by-laws of the Surviving Corporation .	0.128
	(g) As soon as practicable after the Effective Time , each holder as of the Effective Time of any of the shares of MT Common Stock and MT Convertible Preferred Stock to be converted by such holder as elected by such holder as above provided , upon presentation and surrender of such shares to United , shall be entitled to receive in exchange therefor the number of uncertificated , book-entry shares of United Stock pursuant to Section 14-2-626 of the Code and/or cash to which such shareholder shall be entitled according to the terms of this Agreement .	(f) As soon as practicable after the Effective Time , each holder as of the Effective Time of any of the shares of AEB Stock to be converted as above provided , upon presentation and surrender of the certificates for such shares to Fidelity , shall be entitled to receive in exchange therefor the number of uncertificated , book-entry shares of Fidelity Stock pursuant to Section 14-2-626 of the Georgia Code to which such shareholder shall be entitled according to the terms of this Agreement .	0.132
	Each of Company and its Subsidiaries has the corporate or organizational power to own its properties and to carry on its business as now being conducted and as currently proposed to be conducted and is duly qualified to do business and (to the extent applicable in its jurisdiction of organization) is in good standing in each jurisdiction in which it conducts its business , subject in each case to such exceptions as would not have a Company Material Adverse Effect .	(b) The execution of this Agreement and the delivery hereof to the Purchaser and the sale contemplated herein have been , or will be prior to Closing , duly authorized by the Company's Board of Directors and by the Company's stockholders having full power and authority to authorize such actions .	0.133
	Absence of Certain Changes or Events .	Section 3.12 Absence of Certain Changes or Events .	0.138
	Without limiting the foregoing , it is understood that any violation of the foregoing restrictions by the Company's Subsidiaries or Representatives shall be deemed to be a breach of this Section 5.3 by the Company unless such violation is committed without the Knowledge of the Company and the Company uses its reasonable best efforts to promptly cure such violation once the Company is made aware of such violation .	Without limiting the generality of the foregoing , the Company acknowledges and agrees that , in the event any officer , director or financial advisor of the Company takes any action that if taken by the Company would be a breach of this Section 7.11 , the taking of such action by such officer , director or financial advisor shall be deemed to constitute a breach of this Section 7.11 by the Company .	0.144
	Notwithstanding the foregoing provisions of this Section 5.08 , no representation or warranty is made by Parent with respect to information or statements made or incorporated by reference in the Offer Documents which were not supplied by or on behalf of Parent or Merger Sub .	Notwithstanding the foregoing provisions of this Section 5.12 , no representation or warranty is made by Parent with respect to information or statements made or incorporated by reference in the Form S-4 , the Joint Proxy Statement or the Debt Offering Documents which were not supplied by or on behalf of Parent .	0.151

Table 4.2: This table shows examples of deleted sentences with their closest matching sentence found during an iteration of the algorithm. The cosine distance is reported on the far right column. Each example is around .15 cosine distance or below. The examples show that sentences around .15 cosine distance tend to be very similar.

Table 4.3 shows the accuracy for varying cutoff distances. MTD achieves 58.77% accuracy if the cutoff distance is .15. Because a cutoff distance of .15 is somewhat arbitrary, we also include additional higher cutoff distances which, as expected, show an increase in accuracy. As we increase the cutoff distance to .3, the accuracy improves to 73.06%. Based on our observations of sentence pairs, as the distance increases beyond .3, sentences begin to appear very different. We believe using a cutoff distance beyond .3 begins to make less sense. However, it is important to note that occasionally even sentences that are .2 to .4 cosine distance away may appear very similar to a reader because we can recognize things such as different law firm names to be minor differences (e.g., see row 4 in Table 4.1). In addition, although rare, sometimes sentences that are only .2 away can be very different from each other.

$\leq .15$	$\leq .2$	$\leq .3$
58.77%	64.08%	73.06%

Table 4.3: Accuracy for the second test (where the input section and deleted sentence were not included in the similar documents). We consider an iteration to be a hit if there is a sentence with a cosine distance less than or equal to .15 within the top 10 recommended clusters as compared to the deleted sentence. The other two columns show what the accuracy would be if we relaxed our assumptions to .2 and .3 cosine distance.

4.3 User Study

In addition to automatic testing, we created a user study to determine how useful MTD is for people tasked with drafting a contract.[1] With a user study we are able to show that MTD is useful in a practical application. For the user study, we used 30 different sections from merger and acquisition contracts. We used a mixture of different section types including some that are boilerplate (these types of sections do not change much from contract to contract) and some that are highly negotiated between the parties in a contract (meaning that they tend to vary much more than other sections). However, we put a slight emphasis on definitions and termination sections because these types of sections are highly negotiated. Suggestions for these kinds of sections would be of great interest to lawyers because the suggestions would give them a better perspective on how the current contract compares to other contracts. This perspective might be otherwise unattainable without the ability to examine a large corpus of contracts as MTD does.

To perform the user study, we created a website that would show the input section, a group of recommendations from MTD, and a group of recommendations from West. We decided to limit MTD to the first 3 sentences of each of the top 5 clusters because we felt

[1] Approval was obtained from Brigham Young University's Institutional Review Board.

that including more would be too tedious to read for the user study. In practice, an attorney would be able to look at more recommendations from MTD depending on her interest level. We showed the top 20 suggested topics from West. Each algorithm was rated on a scale of 1 to 5, with 1 being not very useful, 3 being hard to tell, and 5 being very useful. The instructions we gave to the users can be found in Appendix B.

We received submissions from 21 users. Each user was a law student who had completed at least a contract class and had some experience during the summer after their first year of law school doing legal internship work. In addition, many of the students had completed a merger and acquisition course, a business organizations course, and other business related courses. To avoid potential bias, each user was shown the documents in random order.

After receiving the submissions, we averaged the scores across all documents. Thus, each document had an average MTD score and an average West score. The average score for MTD was 3.599 and 2.859 for West. We then performed a 1 tailed, paired t-test (using the differences calculated by subtracting the West scores from the MTD scores) to determine whether the MTD scores were greater than the West scores. The p-value was .00028 and thus statistically significant at the .05 level.

4.4 Topic Recommender System Results

Our proposed Topic Recommender System (TRS) was not very helpful for suggesting missing information. As mentioned above, our ideas for locating full sentences that are representative of Latent Dirichlet Allocation (LDA) topics did not appear to be very effective. We found that averaging word vectors in a topic does not seem to be a good technique for finding sentences that represent the topic. We found that the sentences that were returned as the closest matching sentences (smallest cosine distance) were not very similar to the topic. For the most part, the closest matching sentences did not share many words with the topic and did not appear to be closely related to the topic. Although a more elaborate study may be needed to determine why this technique did not work well, one potential reason may be that the words included in a topic did not include common words that are otherwise included in a typical sentence. It is possible that without these common words included, the sentence vectors created for the topics were not accurate representations of a typical sentence within that topic. Because the Missing Text Determiner (MTD) was achieving good results and TRS was not, we used only MTD in the user study and the qualitative comparison with West.

4.5 Neural Translation Model Results

Identifying missing text can be addressed with a translation model. A corpus can be created by removing some amount of text from many different documents. Thus the corpus would consist of input document text (which is artificially missing information) that is paired with its missing text. Thus a model can be trained to predict the missing text based on the text of an input document. We use a preexisting neural machine translation model created by Luong et al. [18]. The model is a sequence to sequence model and uses word embeddings and an attention mechanism.

We trained the Neural Machine Translation model (NMT) [18] using data generated from the corpus of approximately 1200 contracts (containing 97,048 sections) scraped from EDGAR. The training data was generated by randomly removing sentences from the contract sections. Training ran for 9 days on a Titan X pascal architecture GPU.

The results obtain by this model were poor. Only 15% of the sentences were usable, for example: “The Registration Statement will comply as to form with the requirements of the Securities Act and the rules and regulations thereunder.” 45% of the sentences would have required editing to make sense, for example “The table of contents and headings contained herein shall be deemed to be followed by the words without limitation.” In this case it takes a little imagination to make this sentence into something that might be useful. The words following “deemed” make no sense. It would need to be edited to say something like “The table of contents and headings contained herein shall be deemed to have no effect on the interpretation of the agreement.” 40% of the sentences make no sense at all, such as “The parties shall cooperate with the other party to the other party to the transactions contemplated by this Agreement and the transactions contemplated hereby and the transactions contemplated hereby and thereby and the transactions contemplated hereby and thereby and the transactions contemplated hereby shall be effective by the SEC or the other transactions contemplated by this Agreement and the.” Which has no useful meaning and it is hard to even imagine what this might be referring to.

4.6 Qualitative Results Using an Additional Data Set

To determine whether MTD would be effective on other datasets and to further show how MTD is an improvement on West, we tested MTD on the Congressional Record corpus. This allows us to make a direct comparison with West because this corpus was also used by West et al.[20] The data set “consists of all debates from the House of Representatives of 2005” [20]. It was originally created by Thomas et al. [19] We trained siamese CBOW [8] on the dataset and then ran MTD as described above, using Mark Udall’s October 2005 speech (nothing was removed from the speech) as the input document. Like West et al., we did not include text from the debate that Udall’s speech occurred in. Table 4.4 shows the results suggested by West as reported in their paper [20].

Top 20 Topics Suggested by West
Plaintiff
Class Action Fairness Act of 2005
Judiciary
U.S. District Court for the Middle District of Florida
Jury
Will (Law)
Due Process
Trial De Novo
Tort
Advance Health Care Directive
Attorney General
Judge
Supreme Court of the U.S
State Law
Liability
Jurisdiction
Damages
Forum Shopping
U.S. Court of Appeals for the Second Circuit
Product Liability

Table 4.4: A reproduction of the top 20 suggested topics for a U.S. Congress speech on the Lawsuit Abuse Reduction Act of 2005 as reported by West et. al [20]. A dataset of Congressional speeches was used as the background corpus.

Table 4.5 shows MTD results along with topic results reported by West et al. that appear to match the MTD results. The sentences we present all appeared in the top 10 recommended sentence clusters. By comparing the suggested sentences and topics, we can see that the sentences not only contain the topic, but also indicate to a user what should be said about the topic. Sentences provide much more context to a user, allowing them to make more sense of the individual topics contained within the sentences. For example, in row 4 of Table 4.5, we see that a user should possibly be concerned about legislation that overrides beneficial state laws. If the topic "state law" is suggested without context, a user may remain confused about what to say about the topic or why "state law" is an important topic.

In addition, full sentences are able to link multiple topics together, allowing a reader to see how the topics are related to each other. Row 2 of Table 4.5 provides context for the topics "Product Liability" and "Damages." Clearly, product liability cases will almost always involve damages, but by suggesting a full sentence, the user is able to learn (according to one opinion) that the product liability cases and associated damages are becoming overburdensome. By suggesting full sentences, the user has more context and is better able to make a decision about what to write concerning multiple topic combinations.

	MTD Sentence Recommendations	Matching West Topics
1	however , if a court finds that the citizenship of the other class members is not widely dispersed , the opposite balance would be indicated and a federal forum would be favored .	Class Action Fairness Act of 2005, Forum Shopping
2	unfortunately , the food industry has been targeted by a variety of unfounded legal claims which allege businesses should pay monetary damages and be subject to equitable remedies based on novel legal theories of liability for the overconsumption of its legal products .	Liability, Product Liability, Damages
3	the sponsors believe that one of the significant problems posed by multistate class actions in state court is the tendency of some state courts to be less than respectful of the laws of other jurisdictions , applying the law of one state to an entire nationwide controversy and thereby ignoring the distinct and varying state laws that should apply to various claims included in the class , depending upon where they arose .	Jurisdiction, State Law, Class Action Fairness Act of 2005
4	once again , mr. speaker , we have before us a bill that would sweep aside generations of state laws that protect consumers .	State Law
5	it is the sponsors ' intent that although remands of individual claims not meeting the section 1332 jurisdictional amount requirement may take the action below the 100-plaintiff jurisdictional threshold or the $ 5 million jurisdictional amount requirement , those subsequent remands should not extinguish federal diversity jurisdiction over the action as long as the mass action met the various jurisdictional requirements at the time of removal .	Jurisdiction, Plaintiff, Class Action Fairness Act of 2005, Damages
6	encourage the executive branch to follow a doctrine of non-acquiescene by not finding a judicial decision affecting one jurisdiction to be binding on other jurisdictions .	Jurisdiction

Table 4.5: This table shows the power of suggesting full sentences as opposed to traditional topics. The left column contains sentences recommended by MTD and the right column contains suggested topics as reported by West et. al [20]. By suggesting sentences, MTD is not only able to suggest a topic, but it also indicates to a user what should be said about the topic. The recommended sentences all appeared within the top 10 recommended sentence clusters.

Chapter 5

Conclusion

We have presented a new algorithm called Missing Text Determiner (MTD) that is able to make text suggestions to add to a document. MTD takes advantage of sentence vectors, clustering, principal component analysis (PCA), and a background corpus of text to make suggestions. MTD suggestions have been shown to be more useful to users interested in drafting merger and acquisition agreements than the current state of the art topic suggestion algorithm created by West et al. [20]. Our results suggest that providing sentences is more useful than topics for enabling human users to determine what is missing from a text document. Our results also suggest that MTD is more effective than a traditional neural machine translator and the topic recommender system for suggesting missing text. Future work may include adding the sentences to the input document in a way that makes sense with the document overall, or generating new sentences that would be useful for the input text document.

References

[1] Aluminum co. of america v. essex. *F. Supp.*, 499:53, 1980.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. `http://arxiv.org/abs/1409.0473`, Accessed June 7, 2017.

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

[4] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. Fragment-based image completion. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 303–312. ACM, 2003.

[5] Xibin Gao. *Knowledge discovery from business contracts*. North Carolina State University, 2012.

[6] James Hays and Alexei A Efros. Scene completion using millions of photographs. In *ACM Transactions on Graphics (TOG)*, volume 26, page 4. ACM, 2007.

[7] Jeena Joshua and Gopu Darsan. Digital inpainting techniques-a survey. *International Journal of Latest Research in Engineering and Technology (IJLRET)*, 2(1):34–36, 2016. `http://www.ijlret.com/Papers/Vol-2-issue-1/part-2/16-B2016032.pdf`, Accessed January 1, 2018.

[8] Tom Kenter, Alexey Borisov, and Maarten de Rijke. Siamese cbow: Optimizing word embeddings for sentence representations. 2016. `https://arxiv.org/pdf/1606.04640.pdf`, Accessed December 31, 2017.

[9] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. 2015. `https://arxiv.org/pdf/1506.06726.pdf`, Accessed December 31, 2017.

[10] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2001. `http://papers.nips.cc/`

paper/1861-algorithms-for-non-negative-matrix-factorization.pdf, Accessed December 31, 2017.

[11] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015. https://arxiv.org/pdf/1508.04025.pdf, Accessed December 31, 2017.

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf, Accessed January 1, 2018.

[13] Ani Nenkova and Kathleen McKeown. A survey of text summarization techniques. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining text data*, chapter 3, pages 43–76. Springer, 2012.

[14] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[15] 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.

[16] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, volume 14, pages 1532–1543, 2014.

[17] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Matrix factorization and neighbor based algorithms for the netflix prize problem. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, pages 267–274. ACM, 2008.

[18] Rui Zhao Thang Luong, Eugene Brevdo. Neural machine translation (seq2seq) tutorial. 2017. https://github.com/tensorflow/nmt, Accessed January 1, 2018.

[19] Matt Thomas, Bo Pang, and Lillian Lee. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the Conference*

on Empirical Methods in Natural Language Processing, pages 327–335. Association for Computational Linguistics, 2006. URL http://dl.acm.org/citation.cfm?id=1610075.1610122.

[20] Robert West, Doina Precup, and Joelle Pineau. Automatically suggesting topics for augmenting text documents. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 929–938. ACM, 2010.

[21] Albert Au Yeung. Matrix factorization: A simple tutorial and implementation in python. 2010. http://www.quuxlabs.com/blog/2010/09/matrix-factorization-a-simple-tutorial-and-implementation-in-python/, Accessed January 1, 2018.

Appendix A

Regular Expressions for Splitting Sentences

```
def split_up(text):
sentences = re.split(’ \([A-z0-9][A-z0-9]?[A-z0-9]?[A-z0-9]?\) ’, text)
return sentences

def separate_punctuation():
with open(’1200_toronto_style.txt’, ’r’) as in_f:
lines = in_f.readlines()
print "finished reading ", len(lines), " lines"

match1 = re.compile(r’[;|.|(|)|"|:]+\s*$’)
with open(’1200_toronto_style_punc.txt’, ’w+’) as out_f:
for line in lines:
```

Appendix B

User Study Instructions

The following instructions were given to our user study participants.

In this user study, you will be shown various sections of M&A agreement text. For each example, imagine that you are an attorney trying to finish writing the example section. There will be recommendations of things to include in the section from two different algorithms that are intended to help an attorney finish writing the section.

Please rate the suggestions from each algorithm on how helpful, in your opinion, they seem to be for completing the section (1 being not very useful and 5 being very useful).

One algorithm suggests general topics to discuss in the section. It will suggest 20 topics sorted by how strongly the topic is recommended (it considers the first topic to be its best recommendation). The other algorithm suggests specific contract language to add to the section. It recommends groups of sentences. The top recommended group is Tier 1, the second most recommended group is Tier 2, and so on. You may stop at any time and your responses will be recorded. To be eligible for the gift card, please spend 1 hour providing responses. There are 30 sample agreement texts.