EXSTS: EXPLAINABLE SEMANTIC TEXTUAL SIMILARITY

by

Jiarong Cui

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
August 2021

Dedicated to parent, supervisor, teammates, friends and relatives.

Table of Contents

Abstract

We propose eXplainable Semantic Textual Similarity (eXSTS), an application with two visualizations that allows the users to investigate and interrogate the Semantic Textual Similarity (STS) relationship between two documents. eXSTS offers insights for the users who are not familiar with Natural Language Processing (NLP) to the STS relationship between two documents.

eXSTS comprises two parts, the back-end data analysis system and the front-end user interface (UI). The back-end system consists of a Siamese Neural Network (SNN) and $\text{BERT}_{\text{base}}$ pre-trained model. eXSTS receives two documents, and the back-end system splits two documents into sentences, defines pairs of sentences across the two documents, and calculates the STS score of each sentence pair by comparing the sentence embeddings of the two sentences. We normalize all the STS scores and calculate the document STS score of these two documents. The front-end UI visualizes the STS information of sentence pairs through proposed visualizations to explain the sentence pairs that significantly affect the document STS score of two documents and the global distribution of the STS scores of all the sentences pairs across two documents.

eXSTS was invented to deal with the job advertisement classification task. When a job advertisement is entered into eXSTS, eXSTS retrieves the five most relevant National Occupational Classification (NOC) unit group based on the document STS score of the job advertisement and each NOC unit group. The front-end UI of eXSTS demonstrates the STS relationship between the job advertisement and one NOC unit group in the five most relevant NOC unit groups to explore the important STS information across the job advertisement and this NOC unit group and why eXSTS chose this NOC unit group to opt-in the five most relevant NOC unit groups.

Due to lack of ground truth, we invited 20 graduate students to evaluate eXSTS from five perspectives, captured by associated metrics (Efficiency, Effort, Accuracy, Confidence, and Cognitive Workload), and the average score of eXSTS is 4.38 in five metrics.

Acknowledgements

I am eternally grateful to all my family, friends and supervisor for their continuous support in all my endeavours.

Chapter 1

Introduction

Since the 21st century, Artificial Intelligence (AI) has slowly crept into most activity sectors. AI-powered models help IT experts to analyze big data and obtain conclusions or recommendations. When the return result derived from the AI model affects humans' lives, understanding how AI models furnish such results and what factors play an important role in the result-generating process becomes particularly important.

In the last few years, transformer-based language representation models, especially BERT [6] and all its variations [8], achieved state-of-the-art performance across various Natural Language Processing (NLP) tasks, such as Information Retrieval (IR) and Semantic Textual Similarity (STS). The key point to the empirical success of transformer-based models is their huge parametric space and efficient attention mechanism that learns contextual relations between words in a text. Simultaneously, the millions of parameters and the architecture complexity make the transformer-based model be recognized as a "black box." A significant number of researchers are devoted to seeking out the important component that significantly impacts the result of transformer-based models and demonstrates it by visualizations. According to the design logic of transformer-based models, researchers choose attention mechanism as a breakthrough and develop visualizations to offer innovative insights of the result of the data analysis and concise summaries of the important component that lead to this result. Previous studies have uncovered that the attention mechanism can explore the transformer's contextual embedding and provide a natural interface [3, 14] for representing linguistic features and intermediate representations. Moreover, specific attention heads can learn particular dependencies between words and sentences. Based on these advantages of the attention mechanism, existing visualizations [13] have taken a significant step toward extracting and demonstrating the important component of the result-generating process for the user.

STS is basic research that benefits a wide range of text-related applications, like candidate resumes and job advertisement matching. What is STS? In essence, the goal is to compute how 'close' two pieces of text are in meaning (semantic similarity). Visualizing STS presents several challenges.

First, many existing visualizations display the similarity relationship at word-level across two sentences or two documents. Several online applications involving sentence-level visualization can only extract the core sentence from the document. There are no applications that can dynamically display the sentence similarity relationship across two documents.

Second, despite the fact that many papers reveal many unique graphical representations, most existing visualizations focus more on the novelty and sophistication of visualization techniques rather than offering user-friendly visualizations that anyone without relevant expertise can utilize visualization to solve their tasks.

Third, transformer-based models can capture the STS between two documents. However, no application can provide domain experts who have no prior knowledge about NLP with high-value, practical STS component of the result of the AI model.

Research contributions and questions We developed eXSTS, an application that allows the user to investigate and interrogate the important STS relationship across two documents. eXSTS builds on top of the job advertisement classification task. eXSTS receives a job advertisement and retrieves the five most relevant National Occupational Classification(NOC) unit groups and the document STS scores for each NOC unit group. The front-end UI of eXSTS can explore the important STS relationship between this job advertisement and one NOC unit group in the five most relevant NOC unit groups through the proposed visualizations.

The Government of Canada publishes the national reference on occupations to group similar jobs, which is called the NOC that comprises 500 NOC unit groups. NOC unit group can often be linked directly to one or more occupation(s). Each NOC unit group provides a short description of its associated occupation(s), lists its main duties and employment requirements, and provides examples of job titles. The job advertisements with responsibilities, qualifications, and job position introduction come from LinkedIn and Indeed.

In this job advertisement classification task, eXSTS calculates the document STS score between the job advertisement document and each NOC unit group from the NOC pool. According to the document STS score, eXSTS can identify the most relevant NOC unit group and seeks the important STS sentence pair that leads to the document STS score. The front-end user interface (UI) of eXSTS demonstrates the important STS relationship by proposed visualizations and explains the STS relationship across two documents from different perspectives to provide innovative insight into the STS relationship across two documents.

eXSTS provides several insights for the comparisons between sentence pairs across two documents, the important STS relationship of paragraphs across two documents, and the document STS score between two documents. We demonstrate that eXSTS can replicate insights from Hoover et al. [5] and extend it into the document pair comparisons. eXSTS comprises two parts, the back-end data analysis system and the front-end UI. The back-end system consists of the Siamese Neural Network and the $\text{BERT}_{\text{base}}$ pre-trained model. The front-end UI visualizes the important STS relationship across two documents to explain the detailed STS relationship across two documents.

We build eXSTS for the Government of Canada to identify the new job advertisement that does not belong to any NOC unit group. eXSTS helps the Government of Canada to update the existing NOC unit group to capture the new job advertisement. The visualizations in the front-end UI can provide improvement suggestions for the early draft of the new NOC unit group. The human resource of the company can utilize eXSTS to improve their previous job advertisement to match one or more NOC unit groups. The visualizations of eXSTS can provide information about which sentences or paragraphs get matched across two documents and which paragraphs need to improve or rewrite.

eXSTS has been conducted a small-scale user evaluation, in which 20 graduate students interacted with the front-end UI and evaluate eXSTS from five perspectives, captured by associated metrics (Efficiency, Effort, Accuracy, Confidence, and Cognitive Workload). Our main contributions are summarized as follows.

1. eXSTS can analyze the STS relationship across two documents, eXSTS can also discover important STS relationships across two documents at the sentence

level.

2. eXSTS can provide innovative insights for domain experts who may not have a technical background through an intuitive and easy-to-use proposed UI with visualizations.

3. eXSTS allows the user to manually control aspects of the visualization to explore the important STS components across two documents.

The rest of the thesis is organized as follows: In Chapter 2, we survey the related work explaining Semantic Textual Similarity and transformer-based models. Chapter 3 illustrates the basic notions of the visualizations, transformer-based models, and statistical algorithms integrated into the eXSTS. In Chapter 4, we introduce the application scenarios and each part of the front-end UI and back-end system. Chapter 5 shows the user study experimentation process and questionnaires that evaluate every aspect of eXSTS. Finally, Chapter 6 summarizes the conclusion and future research directions.

Chapter 2

Background and Related Work

2.1 Transformer-based Models

The transformer model architecture [12] relies on multiple sequential applications of the self-attention mechanism to draw global dependencies between input and output. The attention mechanism allows the modeling of dependencies without regard to their distance in the input or output sequences within the process. Hence, self-attention layers can reduce the total computational complexity per layer and improve computational performance for a task involving very long sequences. One key factor affecting the ability to learn such dependencies is the length of the paths forward and backward signals have to traverse in the network. Attention mechanisms can shorten these paths between any combination of positions in the input and output sequences to learn long-range dependencies.

With the advantages of the attention mechanism mentioned above, the transformer has been widely used in Natural Language Processing (NLP). In recent years, with further in-depth development of the transformer concept and its extensions into transformer-based language models, achieve higher and higher performance. One of the most widely used language models is BERT [6] (Bidirectional Encoder Representations from Transformers). BERT applies bidirectional training of the transformer and looks at a text sequence either from left to right or combined left-to-right and right-to-left for training. The bidirectionally trained model can have a deeper sense of language context and flow than the transformer (single-direction language model).

Moreover, BERT can be pre-trained on a massive text corpus, and it uses token embedding, segment embedding, and position embedding to compute the semantic representation of an input document. BERT can be fine-tuned with additional layers for specific NLP tasks using a particular dataset. Therefore, BERT is a state-of-the-art model for a wide range of NLP tasks in 2018.

2.2 Visualization Tools

The attention mechanism is used to improve the performance and explain the internal result-generating process of transformer-based models, which motivates a growing body of research investigating the model outputs or the internal vector representation. Researchers first explored how BERT's attention heads behave [3] and found that a large amount of BERT's attention focuses on the separator token [SEP], and attention heads in the same layer tend to behave similarly. In order to report observations about attention behavior and visualize attention, different researchers provide different visualizations [13, 10, 4, 7] from different perspectives. For example, there are three levels of granularity (attention-head view, model view, and neuron view) in Vig's visualization [13]. The neuron view provides two unique insights. The attention scores appear to be largely independent of the content of the input text, and a small number of neuron positions appear to be primarily responsible for the distance decaying attention weight pattern. Sometimes, a visual analysis tool contains several views, like translation view and neighborhood view in Strobelt's visualization [10], which allows for "what if"-style exploration of trained sequence-to-sequence models through each stage of the translation process. Researchers also build the visualization for a specific task. There is a task that needs to analyze attention to help explain the classification decisions made by BERT, like how does one token influence another token across the layers of BERT? Attention Flows [4] adopt the main idea of the sunburst diagram and explore how the model makes decisions via one specific token that it attends to, starting from the classification token at the very last layer and going backward in layers. Attention Flows can support users in querying, tracing, and comparing attention within layers, across layers, and amongst attention heads in BERT.

According to the related paper research, we find that existing visualizations have several issues. Although existing visualizations can provide new insight into the STS component, these visualizations can only demonstrate the STS relationship between two sentences in the same document. Many existing visualizations are built to explain BERT's internal attention structure. These visualizations are not user-friendly for the users who are not familiar with the transformer-based model.

To address the above issues, we developed eXSTS, an application that utilizes a transformer-based model to capture the STS information between two documents and display it to the user through a user-friendly UI. Anyone who does not have related knowledge can interact with the UI to explore the STS relationship across two documents. eXSTS can help the Government of Canada or the human resource team of a company to solve job advertisement classification tasks. eXSTS is a framework that can be adapted to any STS task without substantial coding effort.

Chapter 3

Methodology

eXSTS composes the back-end data analysis system, and the front-end UI. The front-end UI with two visualizations displays the important STS sentence pair across two documents and describes the detailed STS component of it. The back-end system can capture the important STS relationship between two documents and the STS score of sentence pair by analyzing the sentences embedding matrix. Moreover, domain experts who are not IT experts can interact with the front-end UI to explore STS relationship across two documents without understanding the structure of BERT.

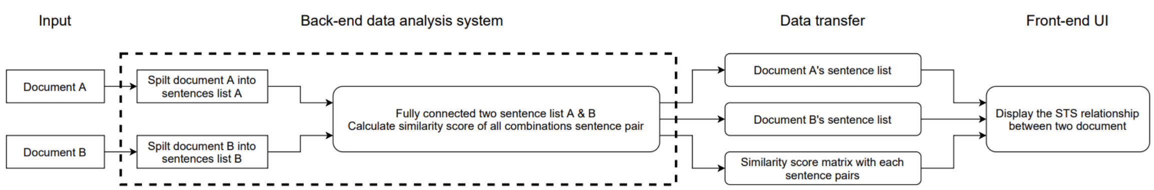

Figure 3.1: **A overview pipeline of eXSTS**. eXSTS receives two documents as input. Two documents are split into sentences, and form sentence lists A and B. Each sentence in sentence list A has fully paired with all sentences in sentence list B, and each combination will be regarded as a sentence pair. Sentence-BERT in the back-end system calculates the STS score of all the sentence pairs across these two sentence lists and obtains a two-dimensional similarity score matrix, which includes each sentence pair's STS score. The back-end system sends sentence lists A, B, and an STS score matrix to the front-end UI, displaying the STS relationship across two documents using proposed interactive tools.

3.1 The Back-end Data Analysis System

Sentence-BERT (SBERT) integrates the Siamese Neural Network (SNN) [2] architecture and BERT to find the important STS sentences in vector space and achieve state-of-the-art performance on STS tasks. We utilize SBERT to calculate the STS score of sentence pairs and identify important sentence pairs across two documents.

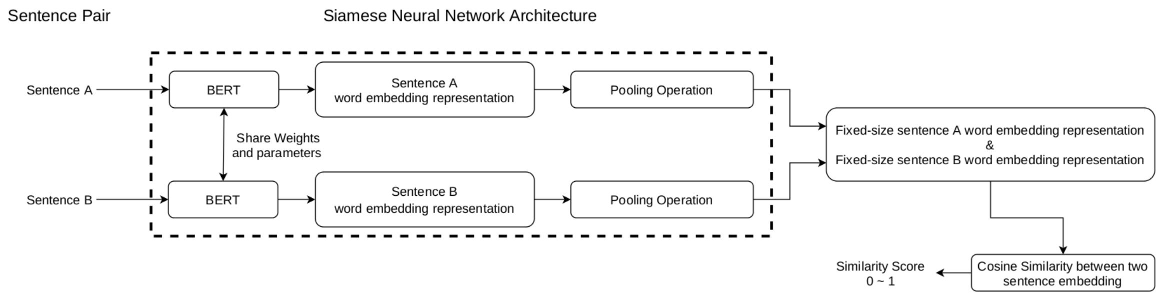

Figure 3.2: **A overview pipeline of the back-end data analysis system**. We add a pooling operation on the top of two BERT sub-networks in Siamese Neural Network architecture to obtain two sentences embedding representation with the same dimensional. The cosine similarity calculate the similarity score between the sentence pair by using the sentence embedding representation. The similarity score from 0 to 1 indicates the relatedness of the two sentences.

3.1.1 Siamese Neural Network Architecture

During the sentence pair analysis process, BERT encodes two sentences into a sentence embedding representation. The contextual semantic similarity of the sentence pair can be computed as the cosine similarity of the embeddings. Negative Manhatten and negative Euclidean distances can also be used as similarity measures to compare the similarity score between two sentence embedding. The experimental results for three measures remained roughly the same in SBERT paper [6]. So, we choose cosine similarity as the only similarity measure.

SNN consists of two identical sub-networks, each capable of learning the hidden representation of an input sentence, which leads to dimensional consistency of sentence embedding representation. The two sub-networks could share the weights and parameters with others, weight-bundle guarantees that their respective networks map the same input to the same locations in feature space. The sub-network is symmetric, and the twin sub-network analyze two sentences. The cosine similarity function obtains two sentences representations with the same dimensional from the sub-network and computes the STS score. The sub-network can be changed based on the tasks and research direction. We use BERT as our sub-network to explain two sentences.

3.1.2 Sentence-BERT

Google AI trained BERT [6] on unlabeled datasets over different sentence scoring tasks. The input of BERT consists of sentence pairs. The first token is always a special classification token [CLS], and sentences are separated by a separator token [SEP]. Thus, BERT can identify two or more sentences based on these special tokens, and the attention mechanism is applied over all input tokens. The [CLS] is encoded, including all representative information of all tokens through the multi-layer encoding procedure. A deep bidirectional representation of BERT is trained by Masked Language Modeling (MLM) technique [11]. Some input tokens are masked at random during the training phase and let BERT predict those masked tokens. MLM stops the target word from seeing itself and prevents the word under focus from actually seeing itself. These techniques feature an essential improvement in context understanding, especially on texts that are "context-heavy." Also, fine-tuning is straightforward and relatively inexpensive. The attention mechanism allows BERT to model many downstream tasks by swapping out the appropriate inputs and outputs, even add a new pooling layer on the top of the BERT. In other words, the in-depth analysis of context and the convenient fine-tuning mechanism are the main reason why BERT is popular and regarded as a revolutionary in NLP.

To adapt BERT for STS tasks and derive fixed-size sentence embedding from BERT, SBERT uses SNN architecture to derive fixed-sized vectors for input sentences. SBERT can produce fixed-size sentence representation (vector space) by adding a pooling operation to the output of BERT. Next, a similarity measure like the cosine similarity function compares two sentences in vector space to identify semantically similar sentences. SBERT is trained on the combination of SNLI [1] and Multi-Genre NLI [15] dataset. This combination dataset contains almost one million sentence pairs, and SBERT is fine-tuned by a 3-way softmax-classifier objective function for one epoch. Based on the evaluation result of SBERT [8] for common STS tasks, SBERT improved the performance for the majority of STS tasks. Therefore, the back-end data analysis system of eXSTS utilizes SBERT to identify important sentence pairs across two documents.

3.2 The Calculation of The Document STS Score

Based on the STS score of all sentence pairs in the two-dimensional STS score matrix, we utilize a mathematical statistic algorithm to calculate the document STS score of two documents. We choose a document pair (NOC unit group "2283 - Information systems testing technicians" and job advertisement "Junior Software Integration and Test Engineer") and check the distribution statistics of all STS scores in the two-dimensional STS score matrix. This documents pair is the most representative of the whole documents, and the distribution statistics of the rest of the documents pairs is even worse.

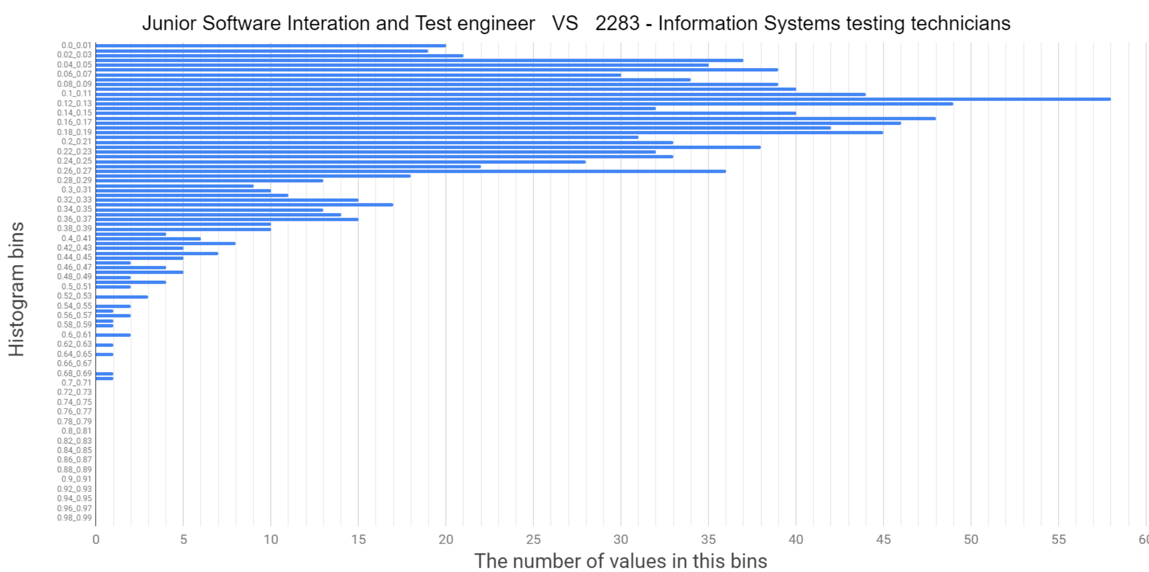

Figure 3.3: **The histogram of STS score in the two-dimensional STS score matrix**. This STS score matrix belongs to the comparison between the job advertisement "Junior Software Integration and Test Engineer" and the "2283 —Information systems testing technicians" NOC unit group. We divided the STS score range(from 0 to 1) into 100 bins. The Y-axis represents the range of each bin, and the X-axis represents the number of STS scores in this bin.

We made the following observations.

1. The highest STS score of sentence pair is in the range of 0.6 to 0.8. However, the proportion of high STS score sentence pairs is no more than 5%. We need to add a normalization process to provide a reasonable STS score of two documents.

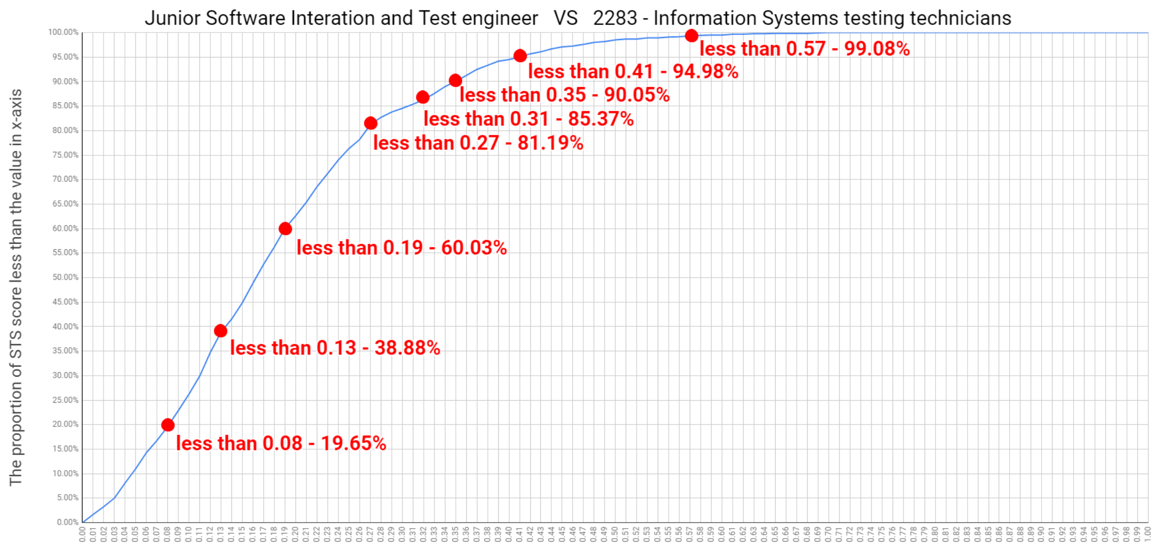

Figure 3.4: **The proportion of STS score distribution**. The X-axis represents the value from 0 to 1, and the Y-axis represents the proportion of STS scores less than the x-axis's value. We add the value of some nodes in the figure. For example, (0.18, 19.65%) nodes represent 19.65% STS score less than 0.18 in the two-dimensional STS score matrix.

2. Most of the STS scores in a two-dimensional STS score matrix are less than 0.4. Because we cannot define a threshold to distinguish between important sentence pairs and the rest by the STS score, we need to add weight for each sentence pair to ensure that every sentence pair across two documents can affect the STS score of two documents and the important sentence pair affects more than the others.

We decided to utilize linear transformation to assign weight to each sentence pair. We split the STS score range of 0 to 1 into X bins and adding weight to each bin.

$$W_1 = 1, W_{i+1} = W_i + 1, W_X = X$$

Where

- W_1 represent the weight of the starting bin. W_X represent the weight of the ending bin.
- The sequence of weight W_i is an arithmetic progression with a common difference of 1. The value of i is from 0 to X.

All the bins are a scale range or an equal interval. The weight was assigned from the weight value 1 of the starting bin to the weight value X of the ending bin. The closer the ceiling of the bin gets to the 1, the higher weight the bin assigned. For example, if the STS score range (0, 1) was split into ten bins, the weight value of the first bin (0, 0.1) is 1, the weight value of the second bin (0.1, 0.2) is 2, and the weight value of the last bin (0.9, 1) is 10.

For each bin i from 0 to X, we can get the weight of each bin W_i, the number of STS score of each bin N_i and the average STS score of each bin AVG_i. Therefore, we can get the summary STS score of each bin.

$$N_i * AVG_i * W_i$$

We added the summary STS score of each bin and divided it by the number of STS scores to get the document STS score of two documents.

$$\frac{\sum_{i=0}^{X} N_i * AVG_i * W_i}{\sum_{i=0}^{X} N_i}$$

For example, if $X = 10$, the formula can be represented by this:

$$\frac{N_{0-0.1} * AVG_{0-0.1} * W_{0-0.1} + ... + N_{0.9-1} * AVG_{0.9-1} * W_{0.9-1}}{N_{0-0.1} + N_{0.1-0.2} + ... + N_{0.9-1}}$$

where

- $N_{0-0.1}$ represent the number of value in the first bin (0, 0.1). $N_{0.1-0.2}$ represent the number of value in the second bin (0.1, 0.2). $N_{0.9-1}$ represent the number of value in the last bin (0.9, 1).

- $AVG_{0-0.1}$ and $AVG_{0.9-1}$ represent the average STS score of the first bin (0, 0.1) and the last bin (0.9, 1).

- $W_{0-0.1}$ represent the weight value of the first bin(0, 0.1) and is equal to 1. $W_{0.9-1}$ represent the weight value of the last bin (0.9, 1) and is equal to 10.

We are experimented with different X values (10, 50, 100, 500, 1000) to find X appropriate in section 5.1. Finally, we choose 100 to become the value of X.

3.3 The Front-end User Interface

The front-end UI needs to demonstrate the comparison between the job advertisement and one NOC unit group. We design two visualizations to demonstrate the STS relationship across two documents from different perspectives.

We develop the relationship visualization to represent the important STS sentence pairs across two documents. The users can change the threshold by using the range slider to control the number of important sentences displayed in the relationship visualization. After identifying the important sentence in the job advertisement, we can seek out the most relevant sentences in the NOC unit group. Relationship visualization connect the important sentence in the job advertisement with the most relevant sentences in the NOC unit group by using lines. The relationship visualization can be regarded as a local visualization. It is centered on a specific sentence in the job advertisement and displays connections to the NOC unit group.

We develop a matrix visualization to represent the global distribution of all the sentences across two documents by using the idea of the heat map. The matrix visualization captures all the connections across two documents. The color variation in the matrix visualization gives obvious visual cues about how the important sentences are clustered over space. The matrix visualization includes two patterns, the entirety and the locality. The entirety displays all the sentence pairs in the matrix visualization. The locality only shows 100 sentence pairs (a specific area in the entirety) in the 10x10 matrix. The users could modify the start of the vertical and horizontal axes to locate the area they want to look at.

3.3.1 Relationship Visualization

There is an visualization called exBERT [5] that provides insights to explore the learned attention weights and contextual representations. exBERT can produce a token embedding of a single sentence and view the representation of curves pointing from each token to every other token. We add several new features on top of exBERT to meet our goal. The primary purpose of relationship visualization is to demonstrate the important sentence pair across two documents.

We identify two main uses that we aim to achieve in relationship visualization:

Figure 3.5: **One-to-Many relationship visualization**. Due to the website limited space, visualizations only display the abbreviation of each important sentence. JOB_22 means the 23rd sentence in the job advertisement, and NOC_15 represents the 16th sentence in the NOC unit group. The user selected a specific sentence (JOB_22) in relationship visualization, eXSTS can display the X (The console can change the value of X) most relevant sentences in the NOC unit group. The opacity of lines represents the STS score between two sentences. We sorted the NOC sentence on the right of the banding descending by the sentence number (e.g., the 15 in NOC_15).

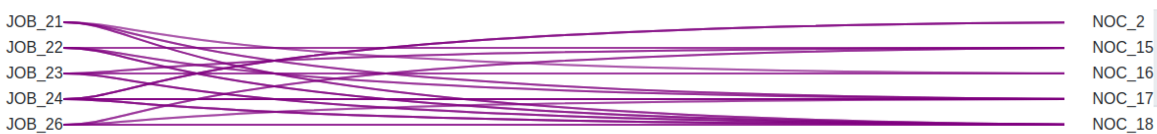

Figure 3.6: **Many-to-Many relationship visualization**. Many-to-Many relationship visualization is assemblies of several One-to-Many relationship visualizations. Based on these five important sentences, we can extract the X most relevant sentences in the NOC unit group and integrate them on the right side of the Many-to-Many relationship visualization. We sorted these five One-to-Many relationships by the sentence number (e.g., the 21 in JOB_21).

Figure 3.7: **The parallel STS relationship**. The users can choose several sentences on the left of the lines to explore the STS relationship of several important sentences in the job advertisement. This figure displays the STS relationship between JOB_21 and JOB_24.

1. **One-to-Many relationship**. When users selected a specific sentence in the job advertisement, for example, the 23rd sentence (JOB_22) of the job advertisement in Fig. 3.5, the relationship visualization can show several relevant sentences in the NOC unit group.

2. **Many-to-Many relationship**. Many-to-Many relationships are assemblies of

multiple One-to-Many relationships. There are five One-to-Many relationships in Fig. 3.6, and we put them all together to represent the parallel STS relationship of the sentence in the job advertisement. The users can click the JOB sentence abbreviation on the left side of the lines to explore the STS relationship of several important sentences in the job advertisement, like the comparison between JOB_21 and JOB_24 in Fig. 3.7.

To draw Many-to-Many relationship, We need to identify the important sentences in the job advertisement on the left side of the relationship visualization. In the two-dimensional STS score matrix, each row means the comparison between a sentence in the job advertisement and each sentence in one NOC unit group. Therefore, we can find the maximum value in a row to represent the important value of this sentence in the job advertisement. We can see sentence's value is greater than the threshold value and regards this sentence as the important sentence in the job advertisement. After we identify the important sentence in the job advertisement, we can extract the X (the console can change the value of X) most relevant sentences in the NOC unit group using the STS score in the corresponding row and display the line and NOC sentence abbreviation (e.g., NOC_2, NOC_15) in the relationship visualization.

Although the relationship visualization can clearly show the important sentence pair across two documents, the relationship visualization has limitations and disadvantages. The relationship visualization is all around the important sentences. Therefore, sentences on the left and right sides of lines may be discontinuous. The document structure has been destroyed. It is challenging for the users to know which part of the document or paragraph gathered several important sentences and got a lot of attention.

3.3.2 Matrix Visualization

To keep the structure of the two documents in the visualization, we borrow the idea of the heatmap to develop the matrix visualization and provide different insights for the STS relationship across two documents and explore the gathering point of the important sentence position in the document.

We identify two main uses that we aim to achieve in the matrix visualization:

1. **The entirety**. The entirety displays all the sentence pairs in the matrix visualization. For example, Fig. 3.8 and Fig. 4.7. The matrix size depends on the number of sentences in the job advertisement and the NOC unit group. The vertical axis represents each sentence in the job advertisement, and the horizontal axis represents each sentence in the NOC unit group. Each cell in the matrix visualization means a sentence pair.

2. **The locality**. In order to zoom into the matrix visualization to see the specific area more clearly, the locality only provide the 10×10 matrix. For example, Fig. 3.9 and Fig. 4.8. The console of matrix visualization allows the user to customize the start of the vertical axis and horizontal axis in the locality. The users could change to any range by using the console of matrix visualization.

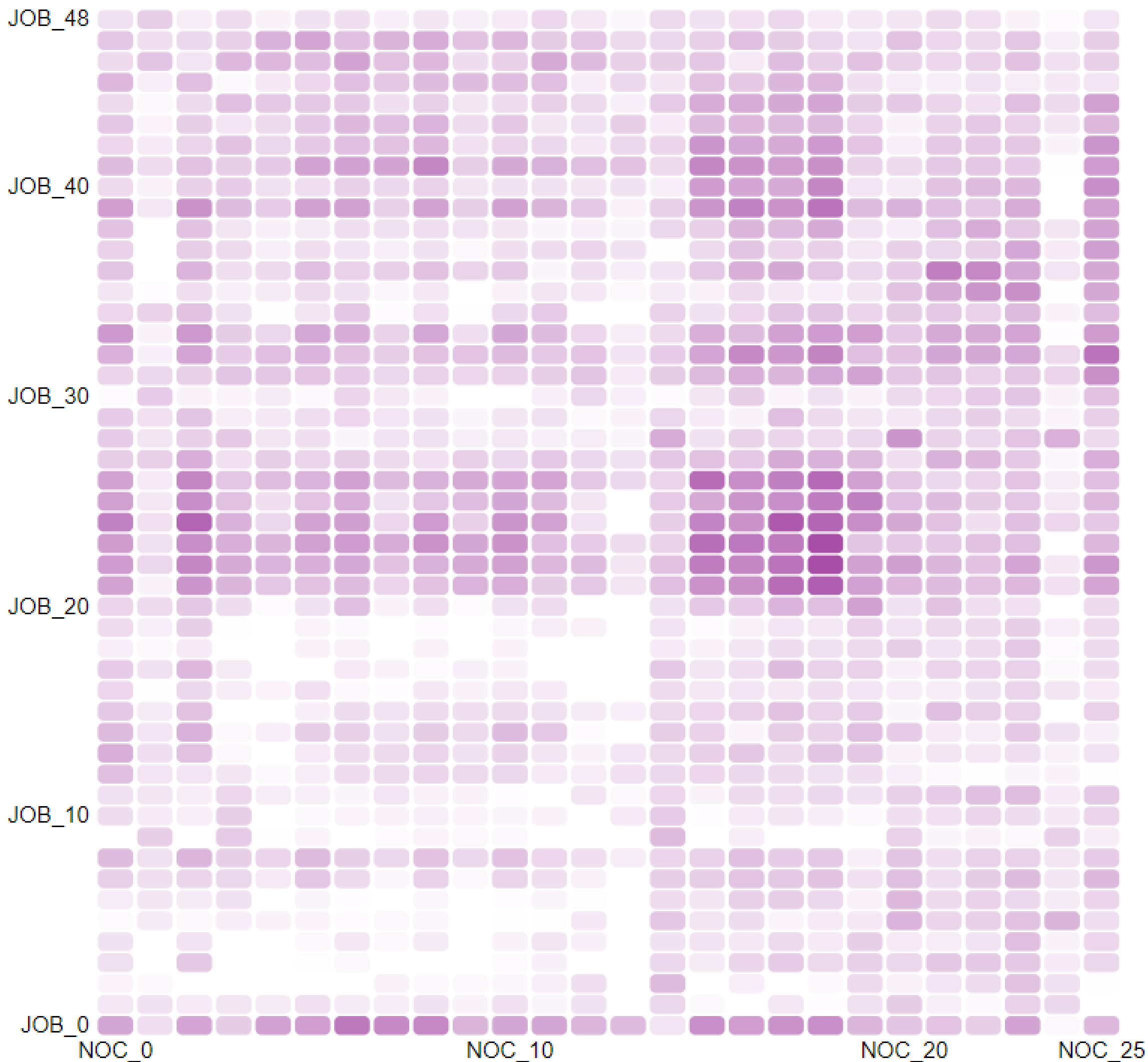

Figure 3.8: **The entirety of matrix visualization**. The entirety of Matrix visualization is the representation of the two-dimensional STS score matrix. The size of the entirety of Matrix visualization depends on the number of sentences in the job advertisement and NOC unit group. The vertical axes represent the sentences in the job advertisement, and the horizontal axes represent the sentences in the NOC unit group. The color scale shows the STS score of the sentence pair. This matrix visualization represents the comparison of job advertisement "Junior Software Integration and Test Engineer" and NOC unit group "2283 - Information systems testing technicians".

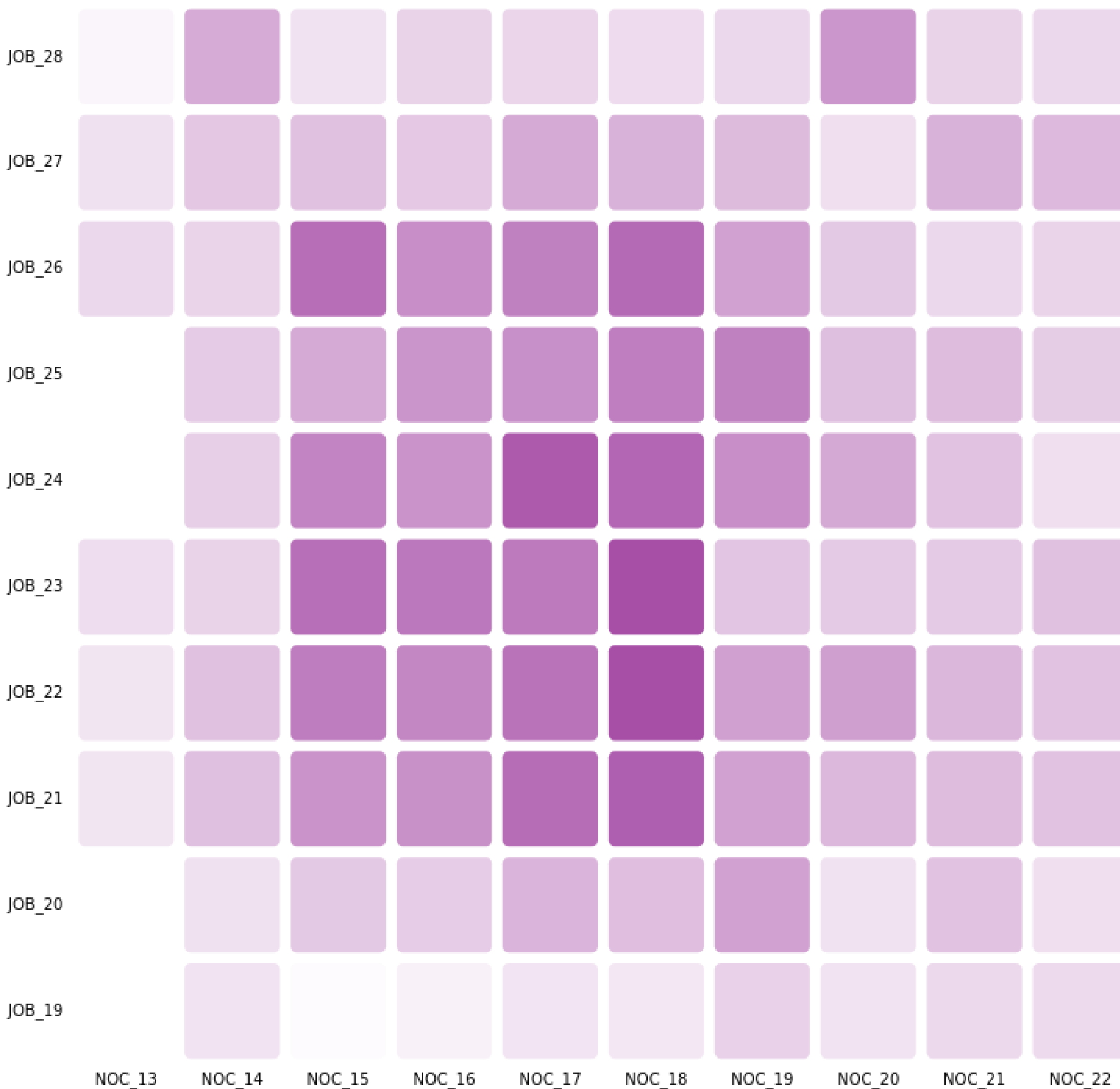

Figure 3.9: **The locality of matrix visualization**. The locality of Matrix visualization helps the users to explore a specific area in the entirety of Matrix visualization. The matrix size of the locality is fixed (10*10). As long as the start of the vertical and horizontal axes is decided, the matrix visualization can display any specific area.

Chapter 4

Implementation

4.1 eXSTS in The Job Advertisement Classification Task

The concrete problem we are solving is the job advertisement classification task. eXSTS receives a job advertisement, and the back-end system needs to retrieve the five most relevant NOC unit groups and the document STS scores for each NOC unit group. The front-end UI demonstrate the important STS relationship between this job advertisement and one NOC unit group in the five most relevant NOC unit group to let domain experts who do not have an IT knowledge background believe the correctness of the retrieved the five most NOC unit groups. eXSTS also provides innovative insights for the detailed STS relationship across two documents.

4.1.1 National Occupational Classification

An occupation is defined as a collection of similar jobs grouped under a standard label for classification purposes. To cluster all job types in society, the Government of Canada publishes the national reference on occupations, which is called the National Occupational Classification[1]. The basic principle of the classification of NOC is the kind of work performed. Job titles are identified and grouped primarily in terms of the work usually performed, determined by the tasks, main duties, employment requirements, and responsibilities associated with each occupation. The NOC comprises about 30,000 job titles gathered into 500 NOC unit groups, organized according to four skill levels and ten broad occupational categories. Each NOC unit group is based on a similarity of skills, defined primarily by functions and employment requirements. NOC unit groups can often be linked directly to one occupation (such as NOC 3113 - Dentists) or more than one occupation (i.e., NOC 2271 - Air pilots, flight engineers, and flying instructors).

[1] NOC Homepage

Each NOC unit group provides a short description of its associated occupation(s), lists its main duties and employment requirements, and provides examples of job titles. However, the Government of Canada did not provide any sample job advertisement or resume. To let domain experts know the job advertisement belong to which NOC unit group precisely and concisely, we use eXSTS to analyze the STS relationship and calculate the document STS score between each NOC unit group in NOC pool and the real-world job advertisement. Based on the document STS score of each NOC unit group, we rank all the NOC unit groups and retrieve the five most relevant NOC unit groups.

4.1.2 Job Advertisement

We extract all kinds of IT job advertisements from LinkedIn and Indeed. Our 200 collected job advertisements comprise job descriptions, employee responsibilities, required skills, qualifications, and duties. During the job advertisement pre-processing phase, we can get the company name in the company description part and use ### to replace the company name in the job description and other parts. We keep some misinformation, like the company description, salary range, and employment type, to test the bias of eXSTS.

4.2 User Interface Design

The front-end user interface consists of the following parts.

The five most relevant NOC unit groups. After the system received a job advertisement, eXSTS retrieve the five most relevant NOC unit groups and display them in the front-end UI. The users can click any NOC unit group in the five most relevant NOC unit groups, Fig. 4.1, and two visualizations show the detailed STS relationship between the job advertisement and the selected NOC unit group.

Job advertisement & NOC unit group content display. Job advertisement & NOC unit group content display the content of a job advertisement (Fig. 4.2) and one NOC unit group (Fig. 4.3). Due to the limited space of the web page, two visualizations can only show the sentence abbreviation. Therefore, when the users click the sentence abbreviation in two visualizations, the front-end UI can highlight the corresponding sentence in the job advertisement or NOC unit group.

Relationship visualization & Matrix visualization. Two visualizations provide the STS relationship across two documents from different perspectives. The console of two visualizations (Fig. 4.5, Fig. 4.6, Fig. 4.7, and Fig. 4.8) allows the user to manually control aspects of the visualization to explore the important STS components across two documents.

The drop-down list. There are several IT fields in the drop-down list. The participants are asked to select their familiar IT field in the drop-down list (Fig. 4.4). Due to the lack of ground truth to evaluate eXSTS, we plan to use the participant's knowledge to evaluate eXSTS from five perspectives. Therefore, the participant should choose the field they know well during the evaluation phase in the user study.

National Occupational Classification Group	Similarity Matching Score
2283 - Information systems testing technicians	46.93%
2173 - Software engineers and designers	43.83%
9222 - Supervisors, electronics manufacturing	43.64%
2147 - Computer engineers (except software engineers and designers)	40.33%
2141 - Industrial and manufacturing engineers	39.0%

Figure 4.1: **The five most relevant NOC unit groups**. After the system received a job advertisement,eXSTS retrieve the five most relevant NOC unit groups and the document STS score of each NOC unit group. The users can click one of the NOC unit groups in the five most relevant NOC unit groups to investigate and interrogate the comparison between a job advertisement and the selected NOC unit group by two visualizations.

Figure 4.2: **Job advertisement content display**. The content display is designed to read the text and observe the sentence position in the document. The sentence abbreviation in two visualizations be feed into here to present content. The selected sentence be highlighted in the job advertisement, and the tooltip display the sentence abbreviation.

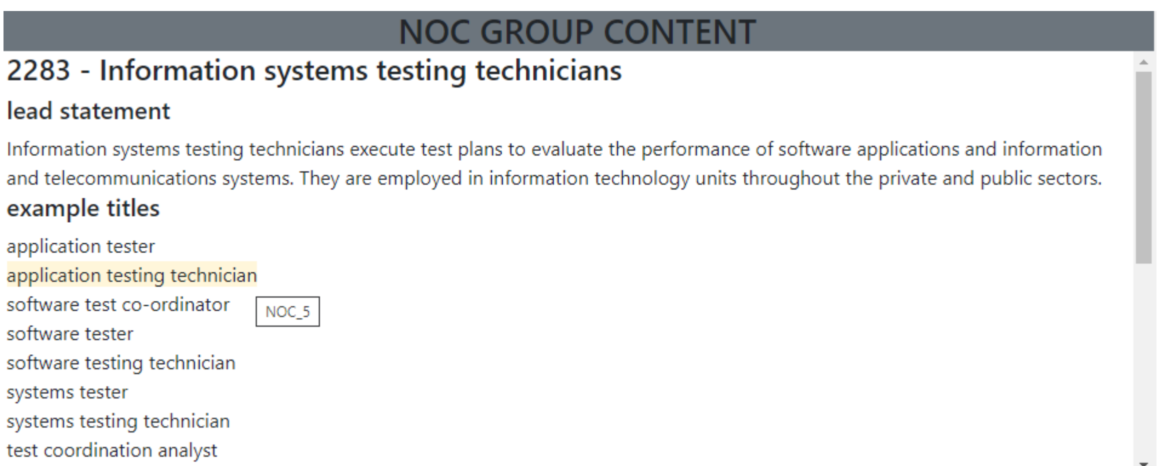

Figure 4.3: **NOC unit group content display**. The content display is designed to read the text and observe the sentence position in the document. The sentence abbreviation in two visualizations be feed into here to present content. The selected sentence be highlighted in the NOC unit group, and the tooltip display the sentence abbreviation.

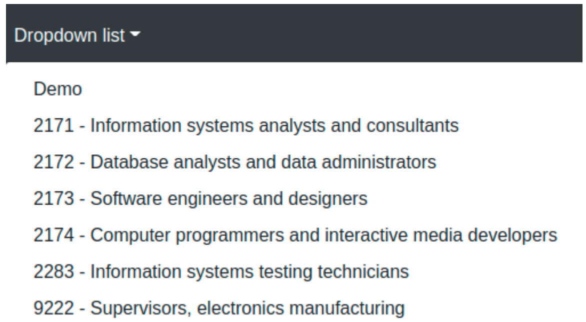

Figure 4.4: **The drop-down list**. The drop-down list in the function bar includes several popular NOC unit groups in the IT field that the Government of Canada mentioned on the website. The users can select their familiar IT field and provide suggestions in the user study's evaluation phase. We use the demo option to introduce our UI functionality in the user study's training phase.

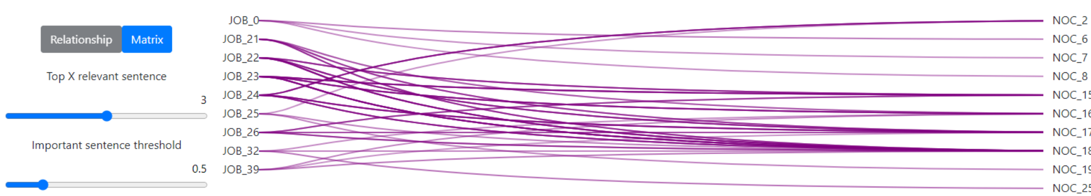

Figure 4.5: **The Many-to-Many relationship visualization with console**. There are two range sliders in the console. The value of the first range slider ("Important sentence threshold") is the threshold to control the number of important sentences on the left of the relationship visualization. The second range slider ("Top X relevant sentence") determine how many relevant sentences in the NOC unit group be displayed in the One-to-Many relationship.

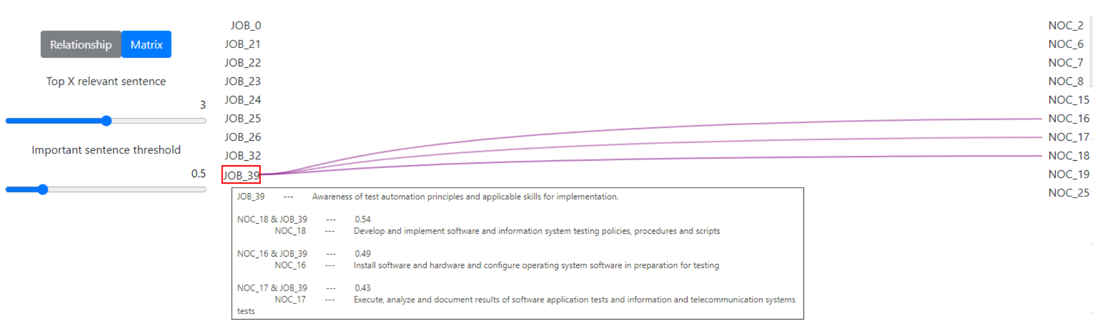

Figure 4.6: **The One-to-Many relationship visualization with console**. The users can click the sentence abbreviation (e.g., JOB_39) to observe the One-to-Many relationship of JOB_39. The tooltip displays the content and STS score of two sentences.

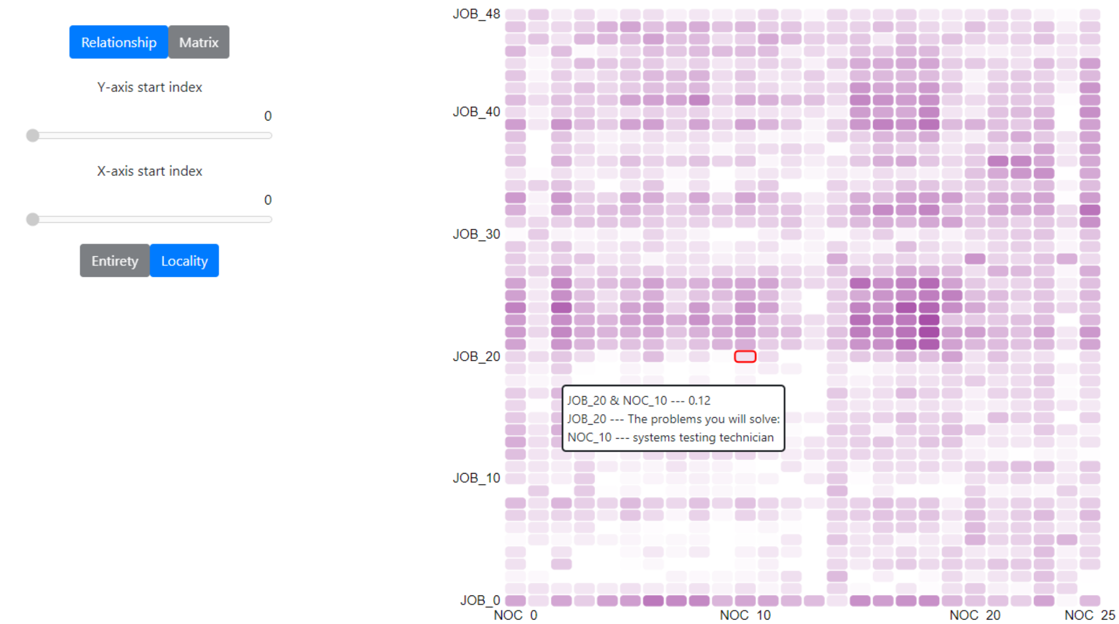

Figure 4.7: **The entirety of matrix visualization with console**. The entirety of matrix visualization demonstrate all the sentence pairs across two documents. The tooltip display the content and STS score of the selected sentence pair.

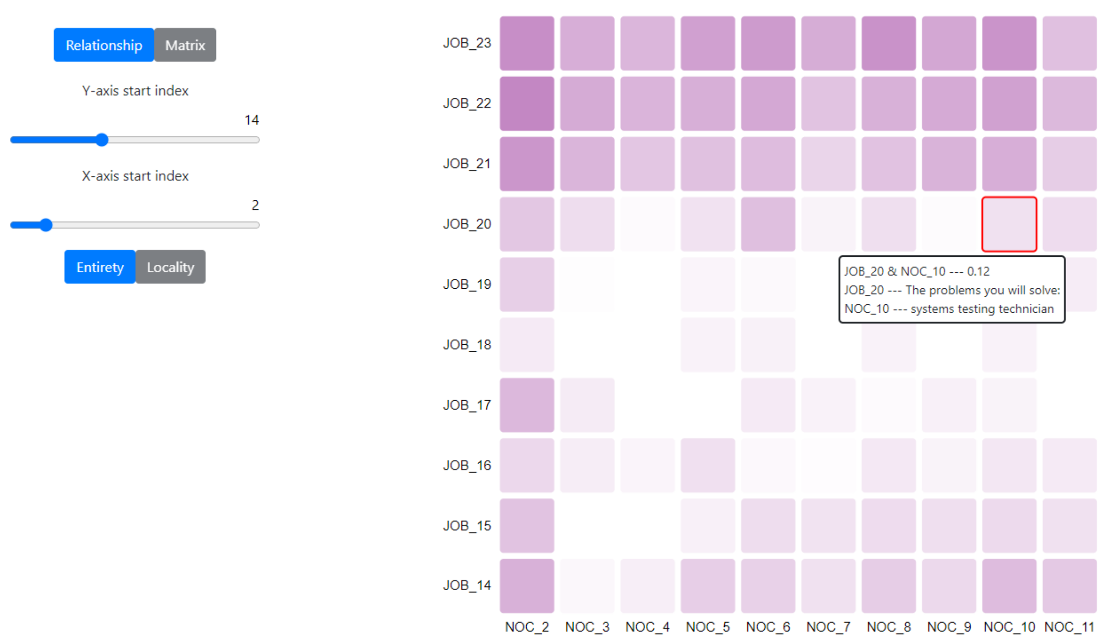

Figure 4.8: **The locality of matrix visualization with console**. The locality of matrix visualization demonstrate a specific area in the entirety of Matrix visualization. Two range sliders control the start of the vertical axis and horizontal axis in the locality of matrix visualization, and the tooltip displays the content and STS score of the selected sentence pair.

Figure 4.9: **The front-end UI with relationship visualization**. The front-end user interface consists of four parts, the five most NOC unit groups, job advertisement & NOC unit group content display, relationship visualization & Matrix visualization, and the drop-down list. The user can click the button to switch between two visualizations.

Figure 4.10: **The front-end UI with matrix visualization**. The console of matrix visualization in the front-end user interface includes two extra buttons to switch between the entirety and locality of matrix visualization.

4.3 Application Scenarios of eXSTS

4.3.1 The Government of Canada

Employment and Social Development Canada and Statistics Canada need to improve the existing NOC unit group or add a new NOC unit group every once in a while. Therefore, the Government of Canada needs to face the following questions when they start working on the updating process:

1. How to identify and extract the new job advertisement that is not involved in the existing NOC unit group?

2. If the Government of Canada decides to improve or rewrite the existing NOC unit group, which NOC unit group needs to be modified? How to modify the existing NOC unit group can achieve a target with the minimum effort?

3. If the Government of Canada decides to create a new NOC unit group, After the employee finishes the early draft of the new NOC unit group, how to improve the content of this new NOC unit group to make it better and better? During the improvement process, which sentences/paragraphs are good enough, and which sentences/paragraphs need to be improved? How to evaluate the final version of the new NOC unit group can perfectly semantic match the new job advertisement?

Based on the document STS score of the five most relevant NOC unit groups, we can filter the new job advertisement that is not involved in the existing NOC unit group. We believe this is the new job advertisement if all the document STS score in the five most relevant NOC unit groups below 15%, even 10%.

If the difference between the document STS score of these 5 NOC unit groups is negligibly tiny and all the document STS score below 10%, none of the NOC unit groups can perfect match this new job advertisement. In this case, the Government of Canada can create a new NOC unit group. After the early draft of the new NOC unit group is entered in the eXSTS, the staff can check whether the new NOC unit group is in the five most relevant NOC unit groups. Matrix visualization can help staff observe which paragraph in the new NOC unit group matched one or more sentences in the new job advertisement. The staff can improve or rewrite the paragraph that did not match the new job advertisement very well.

If the document STS score of the top 1 NOC unit group has a large gap between the other four NOC unit groups, the Government of Canada can improve the content of the top 1 NOC unit group. Matrix visualization can help staff to identify which sentences/paragraphs need to be improved? After the staff improved several specific sentences in the NOC unit group, relationship visualization can check the connection to the new job advertisement.

4.3.2 Company & Ordinary People

The Government of Canada provides some preferential policies to promote the development of some specific industries. Therefore, the content of policies mentions one or more specific job types by using the NOC unit group. The company can utilize eXSTS to modify existing job advertisements or create new ones to match these particular NOC unit groups. The company can enter the new job advertisement into eXSTS to check the five most relevant NOC unit groups. Relationship visualization and matrix visualization observe and report the match condition of the sentence in the new job advertisement.

When ordinary people are looking for a job or following the job stimulus policy, they can use eXSTS to find specific job types that match the NOC unit group in the policy. Furthermore, relationship visualization and matrix visualization explain the STS relationship across job advertisement and NOC unit groups to help people get a job advertisement that perfectly matched the required NOC unit group.

Chapter 5

Experimental Results

5.1 Histogram Bin Size Selection

We choose the most relevant document pair, NOC unit group "2283 - Information systems testing technicians" and job advertisement "Junior Software Integration and Test Engineer" to explain why assign weight to each sentence pair is necessary. We provide the five most relevant NOC unit groups under different bins in table 5.1. We also calculate the average of the STS scores in the two-dimensional STS score matrix and regard it as the document STS score of document pair to retrieve the five most relevant NOC unit groups.

	Average all STS score	10 bins	50 bins
The five most relevant NOC groups	9222 - 17.73% 2173 - 17.34% 2283 - 16.91% 2141 - 16.63% 0211 - 16.60%	2283 - 38.02% 2173 - 34.84% 9222 - 34.51% 2147 - 31.77% 2141 - 30.32%	2283 - 45.27% 2173 - 42.14% 9222 - 41.89% 2147 - 38.70% 2141 - 37.39%
	100 bins	**500 bins**	**1000 bins**
The five most relevant NOC groups	2283 - 46.14% 2173 - 43.03% 9222 - 42.82% 2147 - 39.56% 2141 - 38.23%	2283 - 46.85% 2173 - 43.74% 9222 - 43.55% 2147 - 40.25% 2141 - 38.92%	2283 - 46.94% 2173 - 43.83% 9222 - 43.65% 2147 - 40.33% 2141 - 39.00%

Table 5.1: **The five most score relevant NOC groups of document pair under different bins**. Given the job advertisement "Junior Software Integration and Test Engineer," Each column represents the five most relevant NOC unit groups under different bins. The percentage means the document STS score between job advertisement and this NOC unit group.

We arrive at the following conclusions.

1. The more bins we split, the bigger gap between the NOC unit group candidates in the five most relevant NOC unit groups we can get.

2. The mathematical statistic that we utilized to calculate the document STS score could change the order of the five most relevant NOC unit groups.

3. The more bins we split, the more accuracy is achieved. However, too many bins division does not have much effect on the document STS score. With the number of bins division increased, the calculated quantity rapidly. Finally, we decided to split the STS score range into 100 bins.

5.2 User Study

5.2.1 User Study Hypothesizes

We design a user study. Twenty participants evaluate eXSTS from different perspectives, captured by associated metrics (Efficiency, Effort, Accuracy, Confidence, and Cognitive Workload). We do not have enough resources to find domain experts in other fields involved in our evaluation. Therefore, we collected all kinds of job types in the IT field, looked for participants from the Faculty of Computer Science, and regarded these students as IT experts. We focus on the student who already has several working or interviews experiences to evaluate our eXSTS. When we get enough feedback from IT experts, we extend the application and apply it to other fields.

There are two hypothesizes we want to prove in the user study:

1. **The accuracy of the five most relevant NOC unit groups and the important STS components across two documents**. Given an IT job advertisement, the back-end data analysis system can retrieve the five most relevant NOC unit groups from the 500 NOC unit group pool. In addition, the back-end system can extract the important STS component between IT job advertisement and NOC unit group.

2. **The user-friendliness of the front-end user interface**. The front-end UI can display the STS relationship between two documents clearly and concisely. Furthermore, the users can manually control aspects of the visualization to investigate and interrogate the detailed STS relationship between two documents.

5.2.2 User Study Tasks

There are three phases in the user study of eXSTS, the preparation phase, the training phase and the evaluation phase.

1. In the preparation phase, the participants are asked to read the consent form, and click the "I agree to participate" button before starting the study.

2. In the training phase, the lead researcher provide a live demonstration to explain the functionality of all the elements in the front-end UI. The participants are asked to fill the participants' background survey to help us evaluate the participants' level of NLP knowledge and guarantee the quality of the small-scale user evaluation.

3. In the evaluation phase, the participants select one of the familiar IT fields in the drop-down list. According to the selected options, the related job advertisement be entered into eXSTS, and the front-end display the five most relevant NOC unit groups and the STS relationship between this job advertisement and the most relevant NOC unit group. The participants are asked to interact with the UI to evaluate whether the important STS relationship that the two visualizations provided is reasonable or understandable. The participants are asked to answer the question in two questionnaires to provide their evaluation opinion of eXSTS.

5.2.3 Background Survey

There are 20 graduate students with computer science expertise and working experience involved in our user study. We compiled the participants' backgrounds and asked participants about how familiar participants are with the transformer-based model I used and the STS field. We summarized the corresponding questions in the questionnaire in Appendix A. Based on the result of the background questionnaire. We can conclude that

1. Almost all participants are very familiar with at least one job type in the IT field and have extensive work experience (includes part-time job, full-time job) and interview experience.

2. More than half of the participants with a Master's degree or Ph.D. are engaged in Artificial Intelligence & Machine Learning field.

3. These participants can be regarded as IT experts and trust their feedback of the front-end UI to evaluate and improve eXSTS.

4. Every participant believes online visualization that the user can interact with is one of the best ways to learn and explore new knowledge.

5. Participants with different levels of knowledge about the STS field of study are well distributed.

5.2.4 Result of The eXSTS Evaluation

Based on the background investigation, this user study is a meaningful reference evaluation of eXSTS. Next, we evaluate eXSTS from five perspectives, captured by associated metrics (Efficiency, Effort, Accuracy, Confidence, and Cognitive Workload). Each metric has several questions in the the front-end interactive visualization rating questionnaire and the back-end information accuracy questionnaire (Appendix B). Each question is a statement with five options that correspond to the scale from 1 to 5 (Strongly Disagree $<$ Somewhat Disagree $<$ Neutral $<$ Somewhat Agree $<$ Strongly Agree). There are a number of metrics in visual analytics system to determine the system is useful and usable. We adapt the metrics for human information interaction in the Scholtz's book [9] to our STS research work. Here is the description of each metrics:

- **Efficiency**. Efficiency measures the amount of time the user take to understand each functionality in the front-end UI. Moreover, efficiency is how quickly the user can control visualization to display the specific situation and get the information they want.

- **Effort**. Effort is defined as the total time that the users take to process the information in the front-end UI. Besides, effort measure the amount of time/money the user take to adapt code into their project.

- **Accuracy**. Accuracy represents whether eXSTS can capture the important STS component across two documents to explain the STS relationship of two documents.

- **Confidence**. The user can rely on the STS information or the result that the front-end UI provided as a benchmark/ground-truth to solve their issues.

- **Cognitive workload**. Cognitive workload refers to the extra knowledge/time required of the user to understand the logic, AI algorithm and transformer-based model behind eXSTS.

According to the background survey of participants and the first impressions of participants in the user study online meeting, participants prefer to give a high rating when they fill the questionnaires. Therefore, we improve the 5-point scale in Table 5.2 to evaluate eXSTS.

Score range	Definition
4.5 - 5.0	Excellent
4.0 - 4.5	Good
3.5 - 4.0	Satisfactory
Below 3.5	Failure

Table 5.2: **The 5-point scale**. We use this scale to analyze and conclude the final score of five metrics.

	Efficiency	Effort	Accuracy	Confidence	Cognitive workload
eXSTS	4.30	4.27	4.43	4.46	4.42

Table 5.3: **The final score of five metrics**. We summarized the result of each question in two questionnaires and obtained the score of each metric.

We summarized the result of each metrics and listed the final score in Table 5.3. We can conclude that

1. Although participants thought the functionalities in the two visualizations are helpful and well-integrated, the improvement of the front-end UI is needed, and the front-end UI can be more concise and modern.

2. The important STS component across two documents in the visualizations can help the user understand the STS relationship between two documents.

3. Participants believe eXSTS can be widely used and adapt to the STS tasks. However, the front-end UI includes several professional terms that the user requires more interpretation and time to understand the purpose of each functionality.

5.2.5 Participants Feedback

Participants also provide several valuable feedbacks and comments to point out the concrete problems of the eXSTS. We summed up several feedbacks.

1. Without the functionality interpretations in the training phase, participants have no idea about the purpose of eXSTS, the functionality in the visualization's console, the meaning of the STS component in the front-end UI.

2. eXSTS can only demonstrate the STS relationship across two documents at sentence level. However, IT experts want to know the STS relationship across multiple documents (document-level STS component) or two sentences (word-level STS component).

3. The abbreviation of the sentence (e.g., JOB_1) and the name of the range sliders can be a little confusing at first glance. It may be more clear to replace these titles with a few keywords from each target sentence to see the relationship more clear from a birds-eye-view.

4. The relationship visualization needs more functionalities to provide a better picture of the STS relationship.

5. The website should have a more modern and comfortable layout. It makes a difference.

Chapter 6

Conclusion

In this work, we proposed eXSTS, an application that allows the users to manually control aspects of the visualization to understand the important STS relationship across two documents in the job advertisement classification task.

Based on the frame of existing visualizations, exBERT and heatmap, we proposed two visualizations. The users can look at the important STS relationship and understand the document STS score of document pair from different perspectives. The relationship visualization demonstrates all the important relationships across two documents about an important sentence in the job advertisement. The relationship visualization also displays multiple important sentences in one figure to explore the STS relationship of the parallel sentence. The matrix visualization keeps the sentence sequence to explore the gathering point of the important sentence position in the document. We also conducted a user evaluation, in which 20 graduate students who are familiar with the IT job postings spend one hour with eXSTS, fill the questionnaires and provide feedback.

The individual components integrate many valuable functions, and two visualizations provide essential information to help participants understand the STS relationship across two documents. However, we think only displaying important sentence pairs is not enough. We need several tooltips to explain the details of the generation of the STS score of the document pair. Furthermore, we found it challenging to ensure the user can grasp the point quickly, and the user needs more intuitive and concise conclusions to build users' trust in the STS score.

6.1 Future Work

Future research directions suggested by this thesis are the following.

1. **Add document-level pattern and word-level pattern**. eXSTS only accepted one job advertisement and retrieved the five most relevant NOC unit

groups in the job advertisement task. The document-level pattern of eXSTS is allowed to accept multiple job advertisements and demonstrate the STS relationship across these job advertisements and 500 NOC unit groups. Furthermore, the document-level pattern of eXSTS can adapt into the document clustering field and help users find the document pairs in the extensive document pools. Many existing visualizations use word-level patterns to explain the STS relationship across two sentences, like exBERT. Therefore, we can adapt the existing code in eXSTS to provide the STS component at the word-level.

2. **Extend eXSTS to other domains and using modern layout**. Because of the limited user expertise, eXSTS can only analyze the job advertisement in the IT field. In the future, we can cooperate with companies or the Government of Canada to let other domains experts utilize eXSTS and obtain feedback. This feedback can add more valuable information to help experts master the STS relationship across two documents. According to the investigation, several powerful web frameworks can make our UI concisely and beautifully. In the future, we can improve the layout of the front-end UI and publish eXSTS online to let more ordinary people who do not have high education utilize eXSTS and solve their problems.

3. **Allow user upload two documents**. eXSTS did not allow users to upload their job advertisements and check the five most relevant NOC unit groups. We can develop a new web page with several HTML Textarea elements to let the users enter their job advertisement content and obtain the job advertisement with the standard JSON structure. Next, we can enter the new job advertisement into the eXSTS and analyze the five most relevant NOC unit groups. Furthermore, the users can use proposed visualizations to analyze the STS relationship across any two documents. We can allow users to enter two documents, and eXSTS demonstrate the STS relationship across two given documents.

Bibliography

[1] Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, 2015.

[2] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[3] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, 2019.

[4] Joseph F DeRose, Jiayao Wang, and Matthew Berger. Attention flows: Analyzing and comparing attention mechanisms in language models. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1160–1170, 2020.

[5] Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. exbert: A visual analysis tool to explore learned representations in transformer models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 187–196, 2020.

[6] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[7] Jaesong Lee, Joong-Hwi Shin, and Jun-Seok Kim. Interactive visualization and manipulation of attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 121–126, 2017.

[8] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, 2019.

[9] Jean Scholtz. User-centered evaluation of visual analytics. *Synthesis Lectures on Visualization*, 5(1):i–71, 2017.

[10] Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M Rush. S eq 2s eq-v is: A visual debugging tool for sequence-to-sequence models. *IEEE transactions on visualization and computer graphics*, 25(1):353–363, 2018.

[11] Wilson L Taylor. Cloze procedure: A new tool for measuring readability. *Journalism and Mass Communication Quarterly*, 30(4):415, 1953.

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

[13] Jesse Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, 2019.

[14] Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, 2019.

[15] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*, 2018.

Appendix A

The Participators Background Survey

1. How well do participators know the job type that they have done in the IT field?
    (a) Very good (5 point) - 34%
    (b) Good (4 point) - 52%
    (c) Not Bad (3 point) - 14%
    (d) Bad (2 point) - 0%
    (e) The worst (1 point) - 0%
2. How many IT job interviews have participators had?
    (a) More then three times - 62%
    (b) Three times - 10%
    (c) Twice - 14%
    (d) Once - 5%
    (e) None - 9%
3. How many years of IT work experience do participators have?
    (a) 5 years or more - 29%
    (b) 4 years - 14%
    (c) 3 years - 5%
    (d) 2 years - 19%
    (e) Less than a year - 33%
4. What is the highest level of participators education?

(a) PhD - 19%

(b) Master's degree - 38%

(c) Bachelor's degree - 43%

5. What is participators primary area of study (multiple choice)?

   (a) Artificial Intelligence & Machine Learning - 67%

   (b) Visualization & Graphics - 29%

   (c) Big Data Analytics - 24%

   (d) Algorithms & Bioinformatics - 19%

   (e) Human-Computer Interaction - 19%

   (f) Systems, Networks & Security - 5%

6. In daily life, How often do you use visualizations to solving problems, understand challenging concepts, and analyze data?

   (a) Quite Often (5 point) - 14%

   (b) Slightly often (4 point) - 57%

   (c) Often (3 point) - 29%

   (d) Not Often (2 point) - 0%

   (e) Rarely (1 point) - 0%

7. How do you rate your familiarity with Semantic Textual Similarity, BERT Pretrained AI model, and general visualizations (i.e., LIME, SHAP, EXBERT)?

   (a) Very good (5 point) - 10%

   (b) Good (4 point) - 29%

   (c) Not Bad (3 point) - 29%

   (d) Bad (2 point) - 28%

   (e) The worst (1 point) - 4%

Appendix B

The Front-end Interactive Visualization Rating Questionnaire and The Back-end Information Accuracy Questionnaire

B.1 Efficiency

1. The eXSTS is intuitive to use.
   (a) Strongly Agree (5 point) - 43%
   (b) Somewhat Agree (4 point) - 43%
   (c) Neutral (3 point) - 9%
   (d) Somewhat Disagree (2 point) - 5%
   (e) Strongly Disagree (1 point) - 0%
2. Two visualizations (relationship & matrix) make it easy to understand why these NOC unit groups are in the five most relevant NOC unit group.
   (a) Strongly Agree (5 point) - 62%
   (b) Somewhat Agree (4 point) - 33%
   (c) Neutral (3 point) - 5%
   (d) Somewhat Disagree (2 point) - 0%
   (e) Strongly Disagree (1 point) - 0%
3. eXSTS is a useful way to find the important semantic textual similarity relationship between two documents.
   (a) Strongly Agree (5 point) - 48%
   (b) Somewhat Agree (4 point) - 43%
   (c) Neutral (3 point) - 9%

(d) Somewhat Disagree (2 point) - 0%

(e) Strongly Disagree (1 point) - 0%

4. I found it straightforward to change two range sliders' values in two visualizations to interpret several particular cases.

   (a) Strongly Agree (5 point) - 62%

   (b) Somewhat Agree (4 point) - 29%

   (c) Neutral (3 point) - 9%

   (d) Somewhat Disagree (2 point) - 0%

   (e) Strongly Disagree (1 point) - 0%

5. I think that I would need the support of a technical person to be able to use this system.

   (a) Strongly Agree (5 point) - 43%

   (b) Somewhat Agree (4 point) - 24%

   (c) Neutral (3 point) - 10%

   (d) Somewhat Disagree (2 point) - 13%

   (e) Strongly Disagree (1 point) - 10%

B.2 Effort

1. Two visualization's console feature makes the process of understanding the relationship of sentence pair easily.

   (a) Strongly Agree (5 point) - 67%

   (b) Somewhat Agree (4 point) - 29%

   (c) Neutral (3 point) - 4%

   (d) Somewhat Disagree (2 point) - 0%

   (e) Strongly Disagree (1 point) - 0%

2. I would imagine that most domain experts in other field would learn to use this system very quickly.

    (a) Strongly Agree (5 point) - 33%

    (b) Somewhat Agree (4 point) - 38%

    (c) Neutral (3 point) - 19%

    (d) Somewhat Disagree (2 point) - 5%

    (e) Strongly Disagree (1 point) - 5%

B.3 Accuracy

1. The user cannot clearly understand the semantic similarity relationship because two visualizations have missed some important internal information or framework.

    (a) Strongly Agree (1 point) - 0%

    (b) Somewhat Agree (2 point) - 5%

    (c) Neutral (3 point) - 14%

    (d) Somewhat Disagree (4 point) - 29%

    (e) Strongly Disagree (5 point) - 52%

2. I thought there was too much inconsistency in this system.

    (a) Strongly Agree (1 point) - 0%

    (b) Somewhat Agree (2 point) - 0%

    (c) Neutral (3 point) - 0%

    (d) Somewhat Disagree (4 point) - 43%

    (e) Strongly Disagree (5 point) - 57%

B.4 Confidence

1. I would like to use eXSTS in the future.

    (a) Strongly Agree (5 point) - 81%

    (b) Somewhat Agree (4 point) - 10%

    (c) Neutral (3 point) - 9%

    (d) Somewhat Disagree (2 point) - 0%

    (e) Strongly Disagree (1 point) - 0%

2. I trust the plausible explanation that the application provided, and I felt very confident using the system's retrieval result to solve my problem.

    (a) Strongly Agree (5 point) - 52%

    (b) Somewhat Agree (4 point) - 48%

    (c) Neutral (3 point) - 0%

    (d) Somewhat Disagree (2 point) - 0%

    (e) Strongly Disagree (1 point) - 0%

B.5 Cognitive workload

1. I can obtain most of the knowledge of the semantic textual similarity relationship between two documents, and I did not have to read the Natural Language Processing research paper.

    (a) Strongly Agree (5 point) - 48%

    (b) Somewhat Agree (4 point) - 43%

    (c) Neutral (3 point) - 5%

    (d) Somewhat Disagree (2 point) - 4%

    (e) Strongly Disagree (1 point) - 0%

2. I thought eXSTS was easy to use, and I did not have to learn many things before using this system.

(a) Strongly Agree (5 point) - 72%

(b) Somewhat Agree (4 point) - 14%

(c) Neutral (3 point) - 14%

(d) Somewhat Disagree (2 point) - 0%

(e) Strongly Disagree (1 point) - 0%