Team Translate

Major Freeman, Captain Strickland, Capt Vanderzee

Background/Problem Framing

Captured Enemy Material (CEM) is a vital resource for intelligence gathering and analysis. CEM contains enemy documents, pictures, and other information found by friendly forces. The translation and storage of this information is a burdensome and potentially impossible task for a linguist/intelligence analyst to perform by hand. The development of a tool to increase the turnaround time of translating and searching CEM for critical information is vital to denying the enemy their intended objective. We construct a tool to significantly increase the turnaround time of intelligence analysis and document translation, providing an effective tool to influence today's battlefield.

A mechanism to accomplish the sponsor's objective must focus on document translation and characterization. Most importantly, the end user must be able to easily retrieve the native and translated text of a document. Furthermore, the tool should provide additional information such as similarity among documents and a search function to identify documents containing specific information. Finally, a tool that cannot scale to large quantities of documents (i.e, tens of thousands or more) is useless to an intelligence cell; therefore, a proper data storage platform is imperative.

Utilizing the objectives above, we scope the problem into three different areas: 1) translation/text analysis, 2) data storage, and 3) user interface. Our first area involves the translation of each document from its native language to English utilizing Amazon Translate. Moreover, we compute similarity scores between documents, tag documents as potential duplicates, and provide a list of relevant names found in each document. We then upload the information to a suitable and efficient data storage platform, ensuring the platform is capable of storing mass quantities of information. Finally, we build an applicable user interface to not only visualize the data but also interact with the data to retrieve the desired information.

Translation/Text Analysis

Translation of text documents from various languages to English is a time-consuming task within the Department of Defense (DoD) and Intelligence Community (IC). There are a variety of services with open-source platforms to facilitate rapid translation of text, saving valuable time and resources. Our tool takes advantage of these platforms to establish a workflow to translate CEM to English. Our tool provides intelligence analysts the ability to quickly extract information from documents in minutes vice hours, weeks, or even months. While there are multiple platforms that provide text translation services, our tool utilizes Amazon Translate.

Amazon Translate is a neural machine translation service provided via Amazon Web Service (AWS) to facilitate automated translation of large volumes of text. Presently, the service supports translation between English and the following languages:

- Arabic
- Chinese (Simplified)
- French
- German
- Portuguese
- Spanish

Amazon plans to add Japanese, Russian, Italian, Traditional Chinese, Turkish and Czech to its platform. As with all services provided by AWS, there is a cost. Presently Amazon Translate offers the first 2 million characters translated for free each month. Once this amount is exceeded, the user will pay $15

for every 1 million additional characters translated. There is no minimum requirement and users can take advantage of the service on a pay as you go basis.

While Amazon Translate is a suitable service for translating documents, we found a few limitations. For example, users are limited to 5000 bytes of text per 10 seconds while utilizing Amazon Translate. When exceeded, AWS will throttle usage of the service, slowing down performance. To mitigate the throttling problem, users can separate documents into portions of 5,000 bytes apiece; however, this approach could degrade the quality of the translation. Another limitation within the Amazon Translate service is the user is limited to 10 translation requests per second. For example, if a user submits translation requests for many documents in a short period of time, the use of the service will be throttled. Our best solution to alleviating both throttling issues is to submit a service request to Amazon to increase the throttling thresholds. Amazon granted us 10,000 bytes per 10 seconds and up to 100 transactions per second. While the throttle limit increase boosted the rate at which we processed documents, it still required us to separate large documents into smaller sections to take advantage of the translation service.

The processing routine not only returns translations of the documents; it also computes and returns two measures of similarity between each document and all the others, a measure of the degree of duplication for documents that are highly similar, and a list of found names and possible English transliterations. A cosine distance calculated using all uni–, bi–, tri–, and quad–grams provides the first measure of similarity. The second measure of similarity is based on the names that appear in the document. Names are detected when matched to entries on a provided list, and document name-similarity is calculated using a simple tally of total occurrences of names that are shared across the two documents.

The routine calculates and returns a measure of duplication between two documents that score above 0.9 in cosine similarity. Determining the extent to which one document is a duplicate of another is a non-trivial matter, and the routine uses a relatively expensive O[n*log(n)] algorithm to perform this assessment. The degree to which the second document is a duplicate of the first is calculated as the percentage of the second document constituted by transcribed "chunks" of the first document, divided by the number of distinct chunks. For example, a complete duplicate will be scored as 1.0; a document with 80% of the text lifted straight from another will be scored as 0.8; and a document with two

distinct segments duplicated from segments of the comparison document, summing to an 80% total of its length, will be scored as 0.4.

Finally, the routine returns a list of the names found in the document, as well as their index locations in the text. To facilitate future search queries, the list includes common English transliterations of the found names. These transliterations are added using a provided lookup table.

Data Storage

We employ a MongoDB server to index and make available the document information returned by the above processing routine. To increase the speed by which similar documents are found for any document under consideration, this information is stored within each document's BSON file. As the size of the database grows, this method presents challenges in terms of both storage requirements as well as the 16MB size limit for individual BSON files. An alternative method that retains the similarity information with each document's BSON entry is to store only those IDs for documents whose similarity measure is above a certain threshold. Of course, determining the best threshold level to use may be difficult.

User Interface

We utilize RShiny to construct an applicable user interface for interacting with our database and displaying information to the user. The user interface is deployed on a reverse-proxy server to allow users on government networks accessibility to the tool. Furthermore, the tool is password protected and has direct access with the database.

Our user interface consists of three distinct sections: Database Summary, Search/Document Information, and Visualization. The summary tab produces an overview of what documents are inside

of the database. The size and quantity of documents in the database are displayed to the user along with a treemap illustrating the quantity of document types (i.e, doc, pdf, txt, xls, etc.). The Search/Document Information tab provides the user the ability to interact directly with the documents inside the database. Users are able to query the database for a specific term and a list of documents containing the term are returned to the user. Additionally, the user is able to search for a particular document and return its translation, similar documents, and potential duplicates. The final tab, Visualization, consists of displaying a treemap of word counts for a particular document. This feature allows the user to quickly scan important words in the document to build a description of what the document contains without reading the entirety of the document.

Future Work

While our tool is applicable to the IC, we conclude a training program for properly using the tool is necessary for the tool's success. The tool is not intended to be a decision maker but rather a mechanism to allow analyst the ability to comb through a plethora of information and find what is most useful in a reasonable amount of time/effort. A proper training team to educate the end user of the limitations of the tool is needed for proper employment.

We also see potential scalability issues with the tool's employment. As discussed, throttling issues can severely hamper the tool's ability to provide analysis of CEM back to the user in a short amount of time. While our throttling limitations were increased, we continue experience throttling problems utilizing the translation service by Amazon and recommend continuing to look for other services to mitigate this issue.

Finally, CEM will contain highly sensitive information and must be stored and utilized within a system secure enough to meet the security requirements of DoD. Our workflow does not consider these constraints; therefore, proper authorization to allow the transition of CEM from classified data storage platforms to unclassified should be researched to maintain proper security of the information obtained.

Points of Contact:

Nicholas Freeman (Major USMC)

nicholas.freeman@usmc.mil

Coleman Strickland (Captain USMC)

strickla@usna.edu

Anthony Vanderzee (Captain USMC)

anthony.vanderzee@usmc.mil