



CLARK ATLANTA UNIVERSITY

Annotating Socio-Cultural Structures in Text

Contract No: W913T-12-C-007

Final Report

10/31/2012

Approved for public release; distribution is unlimited.

Table of Contents

Contract No: W913T-12-C-007	0
Final Report	0
Introduction.....	2
Approach.....	2
Phase 1: Manual Annotation of Text Using a Socio-Cultural Taxonomy	2
Interface View.....	8
Lessons Learned.....	18
Phase 2: Automating the Annotation Process	19
Related work	20
Approach.....	22
Results and Validation	27
Integrating the Learning Component into the Workbench	36
Configuring the MaLTAW Learning Component	37
Conclusions and Future Work.....	40
References.....	42
List of Acronyms	44

Introduction

Like many organizations today, the US Army is being overwhelmed by a flood of information. This information has a multiplicity of sources, ranging from intelligence reports, scholarly analysis, images, sensor data, to doctrine documents, and soldier manuals. The ability to model and understand complex urban and social environments is dependent on the ability to extract relevant information from these data sources. This situation does not lend itself to easy solutions; the complexity being driven by the scale and the structure of the information. At one end of the scale is structured information which is amenable to relational database techniques, while at the other end is the completely unstructured information, consisting of text, graphics, video, etc.

Socio-cultural reasoning and ethnographic analysis is increasingly being used by the Army. The understanding of these factors can provide insights into the motivations and thought processes of the adversary. However, much of this information is embedded within unstructured data, particularly text. Extracting this information and transforming it to usable knowledge that facilitates decision making is of tactical and strategic importance. In order to achieve this goal, the relevant inputs have to be generated using available socio-cultural information and using the procedures outlined by doctrine, soldier's manuals and training guides. A necessary step to achieving this goal is the transformation of unstructured data (i.e., text) to semi-structured data. This report details the work performed on Contract No. W913T-12-C-007: Annotating Socio-Cultural Structures in Text by Clark Atlanta University with ERDC-CERL. This contract was initiated December 15, 2011 and concluded on October 31, 2012.

Approach

We consider the transformation of unstructured data towards the support of socio-cultural decision making through a multi-phase approach:

Phase 1: Manual Annotation of Text Using a Socio-Cultural Taxonomy

Text annotation is the practice of marking up text using highlights, comments, footnotes, tags, and links. They may include notes written for a reader's private purposes, as well as shared annotations written for the purposes of collaborative writing, editing, commentary, social sharing, and learning. Annotation of these documents is the first step in the automation of the processing of such documents with applications such as identification of socio-cultural constructs, and improved methods of query and retrieval. Educational research in text annotation has examined the role that text annotations can play in supporting learning goals, communication, and better comprehension. Annotations may also be generated automatically using software tools for web-based collaboration (Koivunen, 2005) or local collaboration (Finlayson, 2011). Folksonomy, a related process, involves creating and managing tags to annotate and categorize text. Tags are assigned in a collaborative process across many users who contribute to the same corpus. Here the set of tags is dynamic and not predefined. Keyword assignment (subject indexing) using a controlled vocabulary of keywords is another related task. The association of keywords with documents is done in a controlled fashion (unlike tags in a folksonomy). Keywords are usually intended to facilitate content-based search rather than category-based browsing which is appropriate for a labeling context. A

thesaurus or external hyper-linked structure such as Wikipedia is often employed to cope with the different task of keyword assignment.

ERDC-CERL provided the CAU team with a text corpus of 473 documents. These documents, mainly in the PDF format were grouped into eleven loose categories. The documents covered a wide variety of topics, from Agriculture to Psy-ops, and were reports on US Army efforts across a wide geographical region. Figure 1 shows a view of the document set by category.

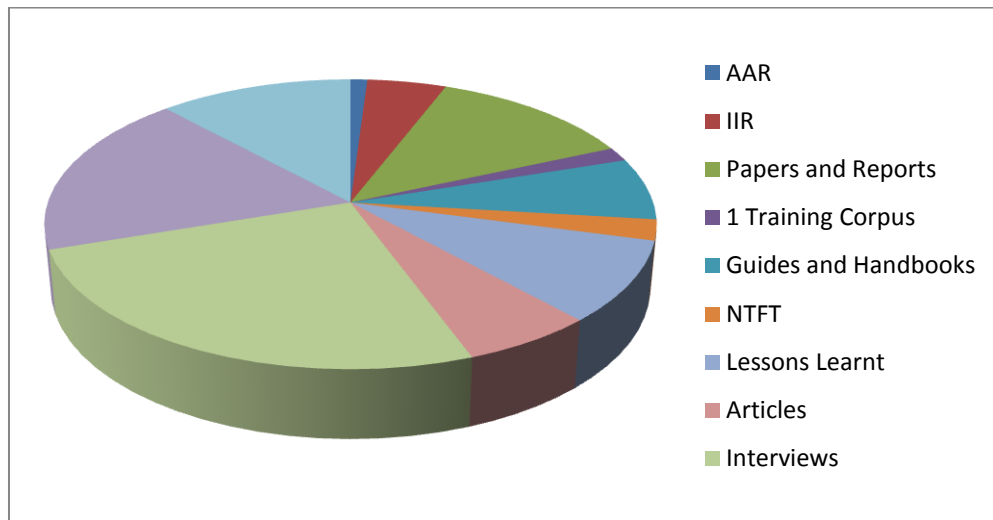


Figure 1: Numbers of Documents by Category

ERDC-CERL also provided a taxonomy for the annotation of the documents. The two-level taxonomy is based on an abstract characterization of phrase type (Level 1), which may be specialized by a Level 2 descriptor. The Level 1 and Level 2 of the taxonomy provided by ERDC-CERL are shown in Table 1 and Table 2 respectively. This taxonomy is used to annotate the documents provided.

L1 Taxonomy
entity/agents entity/events entity/info entity/institutions entity/materials entity/organizations entity/physical_behaviors entity/physical_infrastructures entity/places entity/services entity/social_behaviors entity/social_infrastructures entity/technical_capabilities entity/time

Table 1. Level 1 Taxonomy

L2 Taxonomy		
administrative agreement agriculture authority civilian communication conditions conflicting contractor criminal definition dislocated economy education environment extremism food	global governance health Illicit indigenous labor language liaison licit local_governance military negotiation oversight perspective pets political	private psychological public public opinion purpose relationships relief religion requirements return routine security social transition transportation utilities

Table 2. Level 2 Taxonomy

Considering that the text corpus provided by ERDC-CERL was large, the CAU team selected a subset of the documents for manual annotation. The selection criteria for the documents were the following in descending order of relevance:

- Subject matter primarily relating to socio-cultural events and activities
- Geographical representation
- ERDC-CERL categories

The selection methodology is of necessity abstract, since most of the documents do not neatly fit into socio-cultural or geographical categories except for the ERDC-CERL categories, which reflect the type of document. Fifty nine documents were chosen for manual annotation. Figure 2 shows the breakdown of the documents selected by ERDC-CERL category representation. Table 3 indicates the titles of the documents selected.

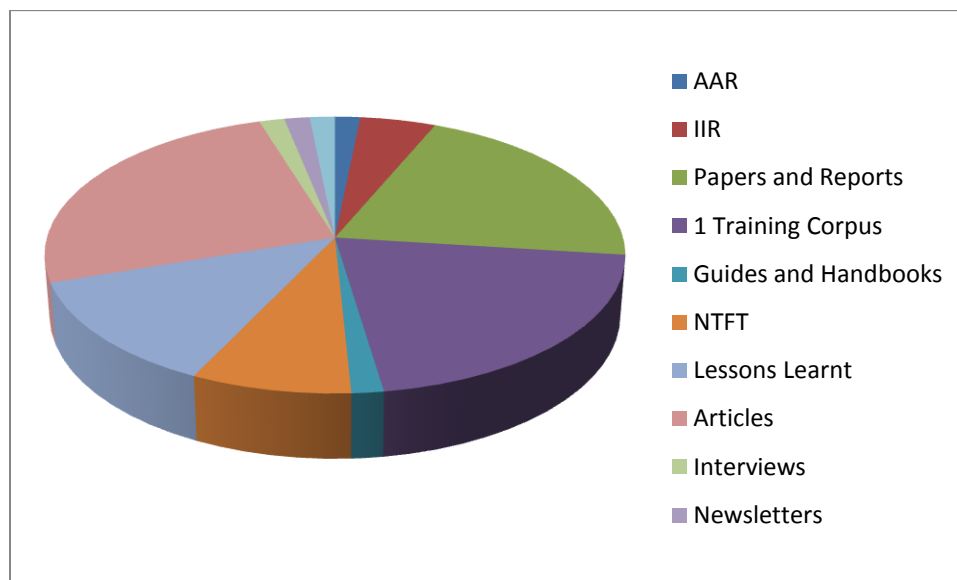


Figure 3: Subset of the Text Corpus Selected for Manual Annotation

Chickens
COIN in Philippines
Connecting to the Populace
Consequence Management
Counterterrorism in the Sahel
Effective Use of HTT
Empowering the Shura for IO
Engaging Women on the Frontline
Fallacy of COIN
Going Far Softly
Green interview
HTT Debrief
Impact of Cultural Awareness
Importance of Politics in COIN
Improvements for Iraqi Medical Clinics
Integrating Latin American Militaries
Integration of SF and USAID in Afghanistan
Intelligence Support to PsyOps
Interagency Response to Natural Disaster
IO From Good to Great
Iraq Building Civil-Military Capacity
Iraqi Funerals
Local Governance and COIN in Afghanistan
Marine CA Detachment Operations
Marines Invest in Afghan Projects
Medical CA1
Mullah Engagement Program
Multinational Corps in Iraq
mylist.txt
NFTF Reeves Mosul Dam
NFTF Strategic Religious Advisement in Iraq
Nonlethal Targeting
One Elder at a Time
Organizing, Staffing, and Focusing Non-Lethal Fires
Pakistan Earthquake Relief Ops
PsyOps in Haiti
Rebuilding Agriculture in Afghanistan
Relationships Matter
School Partnership Program
Shaping Afghani Battlefield in Dari and Pashto
Three Cups of Tea and an IED
Throwing Rocks
Training Aims to Deter Extremists in Africa
USAID Projects
Village Stability Methodology
Water Irrigation Techniques in Iraq
Winning the War and the Relationships
Writing a UCP 2
Youth Shuras Address Education

Table 3: Titles of Manually Annotated Documents Selected

The documents provided were in the PDF format with embedded links, graphics, photographs and meta-data (headers, foot notes, page markings).

Document Preprocessing

The documents are first converted to plain text using **Adobe Acrobat**, and the open source PDF tool, the **Foxit Reader**. As a second step the text documents are cleaned and validated removing duplications or other anomalies introduced by the conversion process. A standard text editor, **TextPad** was used for this step. The next step of of this process is the conversion of the text document to the eXtensible Markup Language (XML) and the identification of the parts of speech (POS) within text, using the Stanford Part of Speech Tagger (Stanford Log-Linear, 2011).

The ERDC-CERL taxonomy is then used to label the tasks and sociocultural information in the training corpus. A software tool, the Machine Learning for Text Annotation Workbench (MaLTAW) was developed to ease the difficulty of manual annotation. A total of fifty nine documents were manually annotated, and a software utility to aid the annotation was developed during Phase 1.

Using MaLTAW for Manual Annotation

MaLTAW is executed by double clicking the icon for TextAnnotator (Application) in the Version 5.4.1 directory. This execution brings up the MaLTAW interface seen in Figure 4.

Interface View

Figure 4 shows an abstract view of the MaLTAW interface. In the following discussion we will expand on the use of each Pane within the context of annotating text.

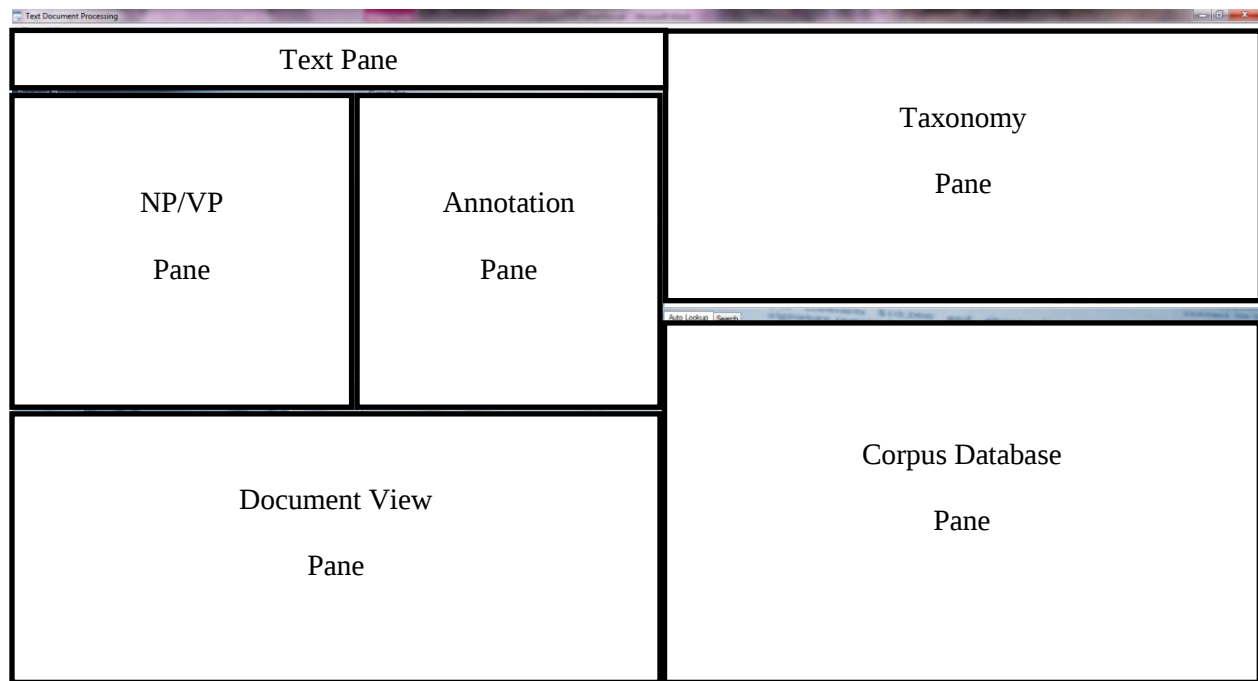


Figure 4: Components of the MaLTAW Interface

Brief descriptions of each pane are given below:

Text Pane: Indicates the sentence currently being annotated

NP/VP Pane: Shows the sentence parsed using the Parts of Speech tagger

Document View Pane: Specifies the document (being annotated) in three different views- as a collection of phrases (Noun Phrases or Verb Phrases), as an XML file, or as a Text File.

Annotation Pane: Shows the current annotation (if available) and system suggested annotations

Corpus Database Pane: Shows the annotation for exact or similar phrases that are already in the database. These phrases are extracted from previously annotated documents and inserted into the database.

Taxonomy Pane: Specifies the taxonomy used to annotate the document. In the current application we use the Level 1, Level 2 taxonomy. New concepts may be added to or deleted from the taxonomy.

Interface Detailed View

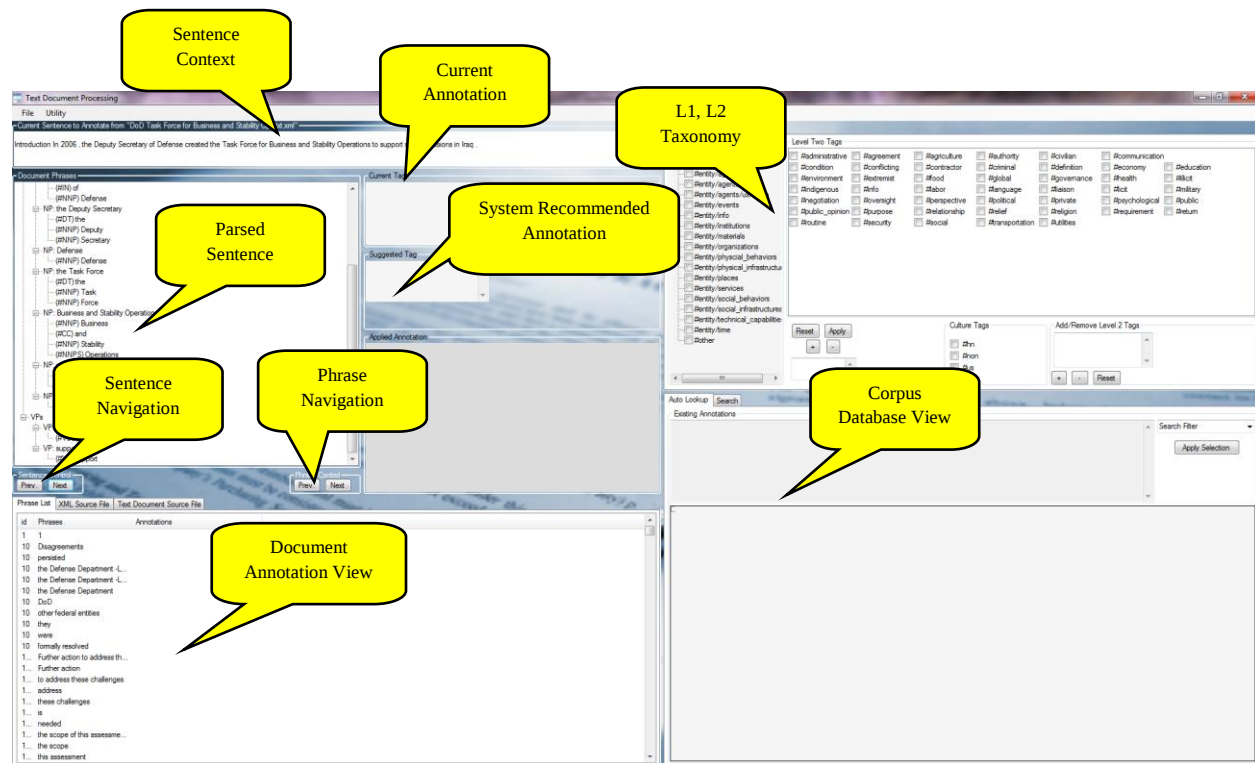


Figure 5: Detailed Functions of the Interface

NP/VP Pane:

- *Sentence Navigation* : Allows the user to navigate the document, sentence by sentence
- *Phrase Navigation* : Permits navigation of through each sentence, one phrase at a time

Annotation Pane:

- *Current Annotation* : Shows the annotation of the current phrase (if one exists)
- *Suggested Annotation* : Displays the system suggested annotation

Step 1: Pre-processing the Document in MaLTAW

- The text document to be annotated has to be first parsed using the Stanford Parts of Speech tagger and converted to an XML document both components which are done through the Import function of MaLTAW. Documents may be imported in a batch mode (multiple) or singly.

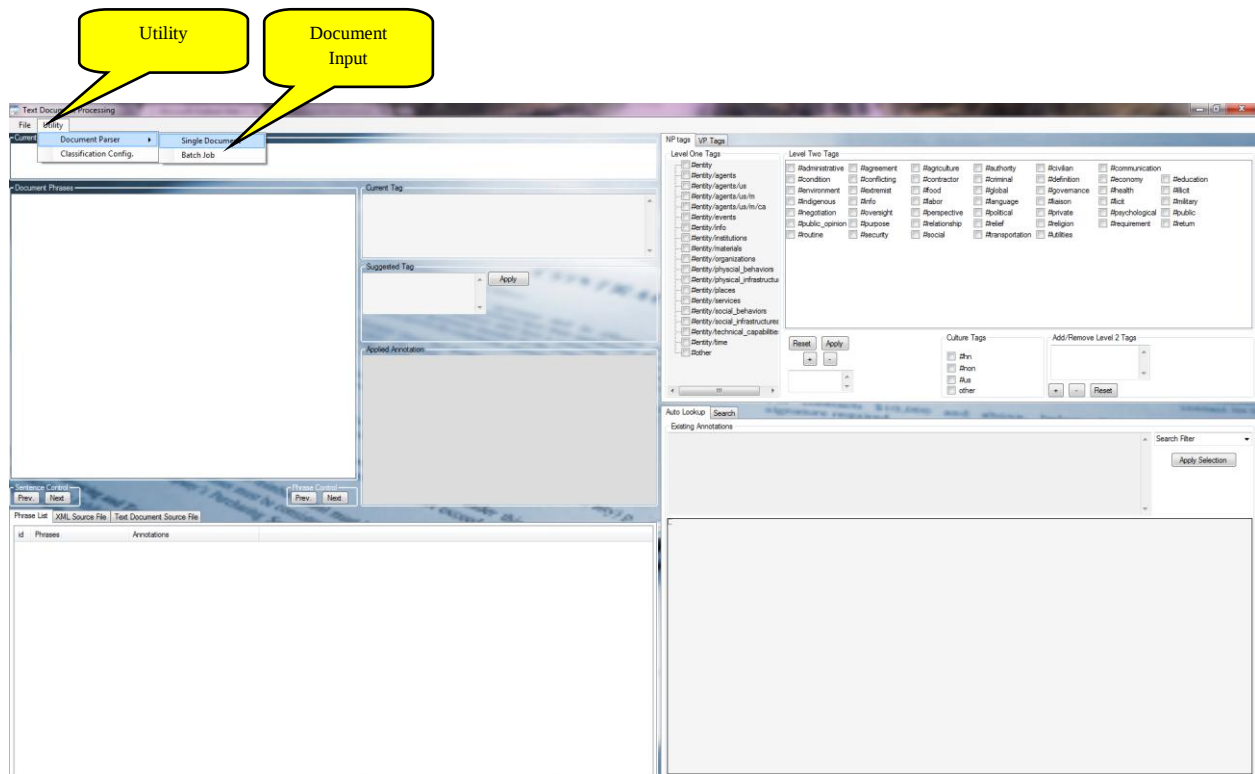


Figure 6: Import Text File(s) into the MaLTAW System

Step 2: Loading Pre-processed Document

- Pre-processed POS tagged XML documents may be loaded into the MaLTAW system using the **File/Open** option.

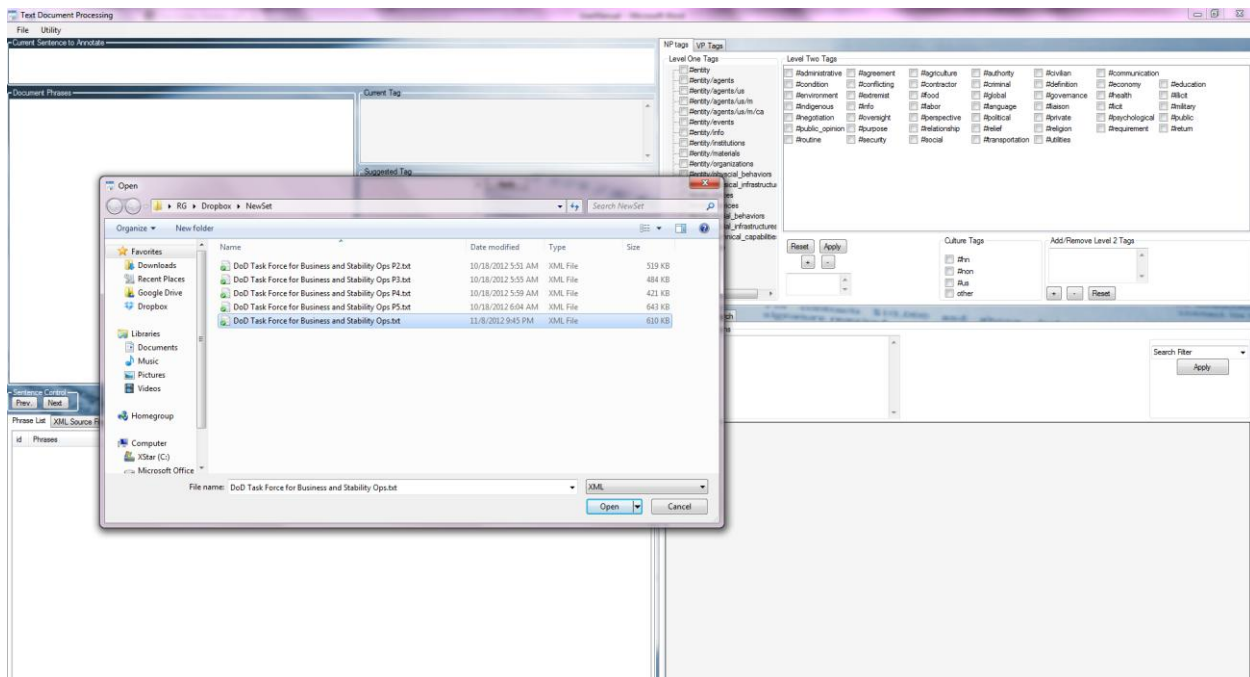


Figure 7: File/Open to Load File to be Annotated

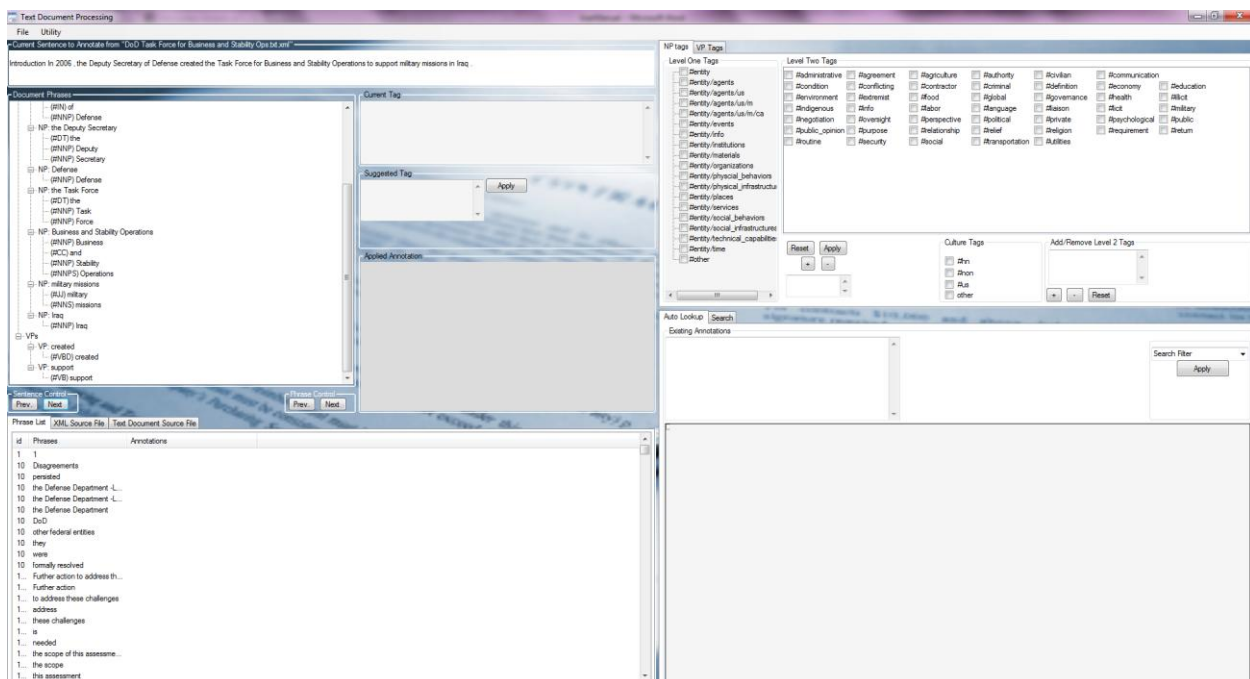


Figure 8: Loaded File (to be Annotated)

Step 3: Annotating the Document

- Each sentence is parsed into Noun Phrases and Verb Phrases. The User Interface provides a mechanism to move between sentences in the document, between phrases in a sentence, and between Noun Phrases and Verb Phrases.
- MaLTAW generates a composite suggested annotation based on the Learning Component and a Phrase Match
- The user may choose to accept the MaLTAW suggestion (using Apply) or choose to annotate the phrase directly using the Taxonomy Pane.
- The Corpus Database shows how the same or similar phrases have been annotated in other documents. The Sentence Context shows the sentence within the Document where the similar annotation has been applied.

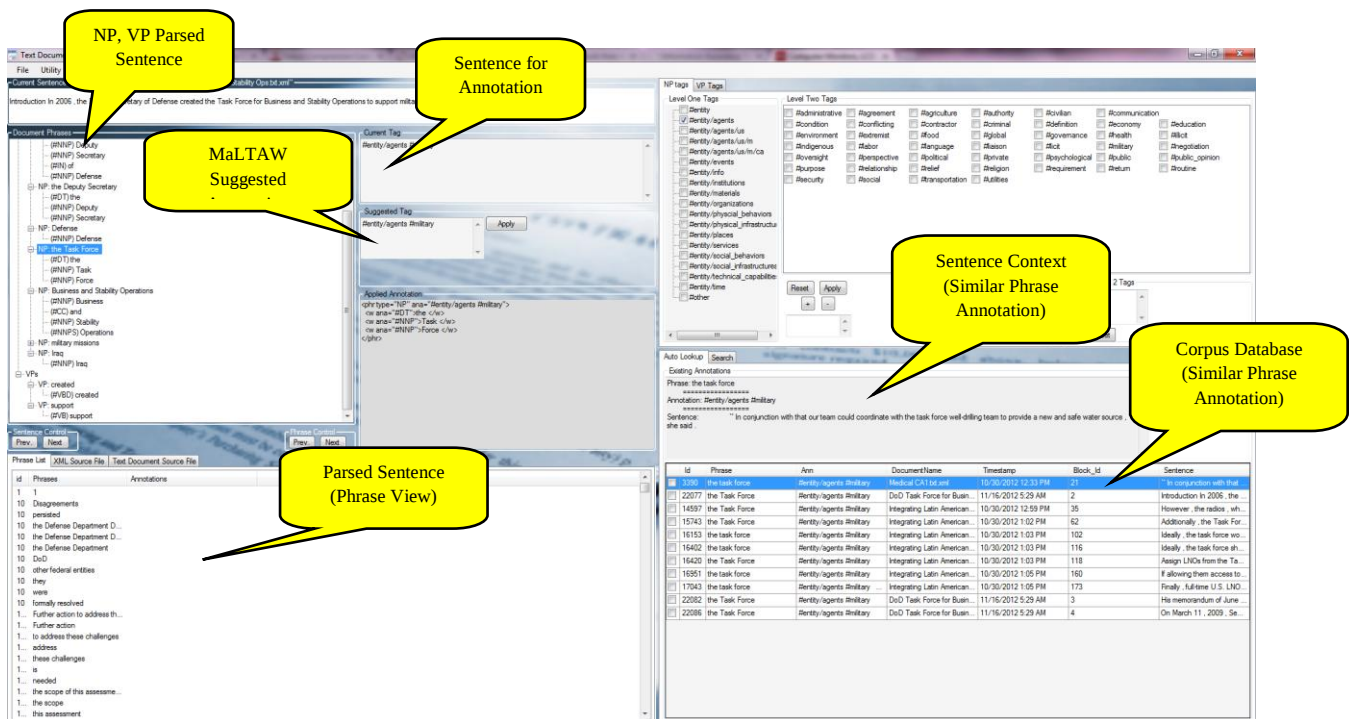


Figure 9: Annotating the Phrase Options

In the example, the sentence to be annotated is:

“In 2006 , the Deputy Secretary of Defense created the Task Force for Business and Stability Operations to support military missions in Iraq .”

from the document *DoD Task Force for Business and Stability Ops.*

The sentence is parsed into Noun Phrases and Verb Phrases. We choose to annotate the Noun Phrase “the Task Force”, which has been parsed as shown in the Figure 10.

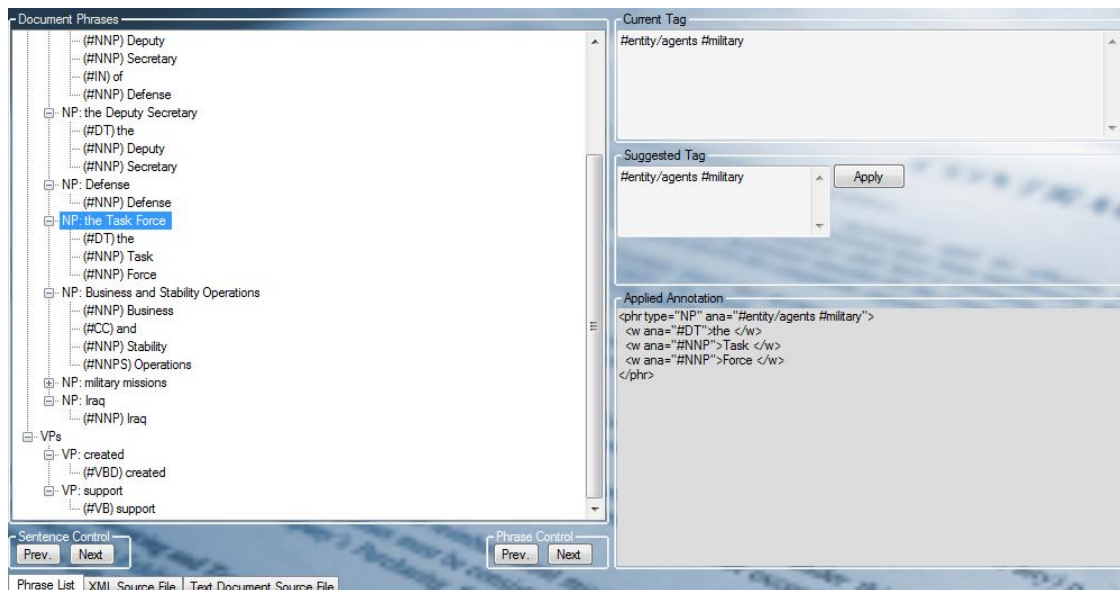


Figure 10: Expanded View of Sentence/Annotation Pane

The user has two choices at this point. They could either **Apply** the “Suggested Tag” option or annotate the Phrase themselves using the Taxonomy Pane. Using the Taxonomy pane requires a selection of the Level 1 and Level 2 tags as shown in Figures .

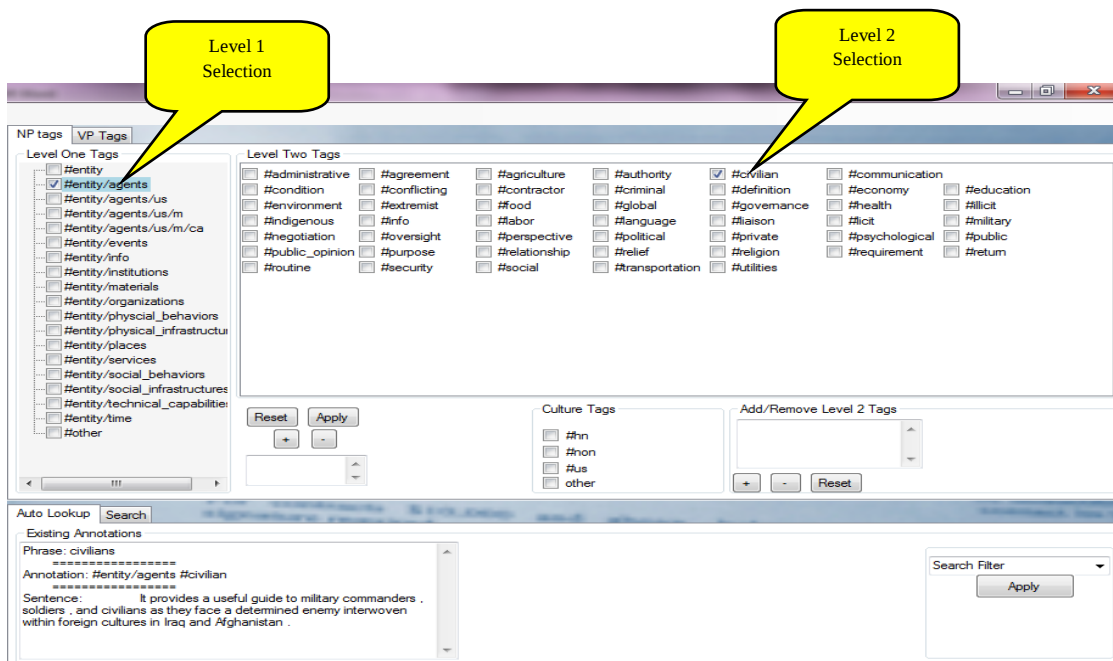


Figure 11: Manual Annotation of Phrase

Using the Corpus Database

The Corpus Database provides an alternate mechanism for aiding the annotation. It searches through the annotation database to find matches to the word/phrase being annotated. The match could be exact or approximate. The user may apply the annotation retrieved from the database to the phrase.

The example below shows the result of the retrieval from the corpus for the phrase “NGO’s”. This phrase has been annotated as **#entity/agents #entity/organizations #private**. Additional information is provided in this pane- the source sentence and source document are indicated. This may be used to determine the context of the phrase and thereby choose the most appropriate annotation. The Corpus Database may be used as a standardization and quality control mechanism for the annotation process.

The screenshot shows a software interface for searching a corpus database. At the top, there are tabs for 'Auto Lookup' and 'Search'. Below the 'Search' tab, the 'Existing Annotations' section displays the search phrase 'NGOs' and the resulting annotation: '#entity/agents #entity/organizations #private #global'. A 'Sentence' field shows a snippet from a document about the XVIII Airborne Corps. To the right, there is a dropdown menu set to 'Exact Match' and an 'Apply Selection' button. Below this, a table lists search results with columns for Id, Phrase, Ann, DocumentName, Timestamp, Block_Id, and Sentence. The first result (Id 3613) is highlighted in blue.

Id	Phrase	Ann	DocumentName	Timestamp	Block_Id	Sentence
3613	NGOs	#entity/agents #entity/or...	Building Civil Capacity in Ir...	10/30/2012 12:33 PM	13	XVIII Airborne Corps and i...
4332	NGOs	#entity/organizations #pri...	Connecting to the Popula...	10/30/2012 12:35 PM	25	The district chief , in tum ...
6758	NGOs	#entity/organizations #pri...	PakistanEarthquakeRelief...	10/30/2012 12:40 PM	18	Lifeline included military p...
9607	NGOs	#entity/agents #global #...	PakistanEarthquakeRelief...	10/30/2012 12:46 PM	102	He bent every rule in the ...
20173	NGOs	#entity/agents #entity/or...	CA in East Africa.txt.xml	11/5/2012 12:55 PM	107	A crucial component in ef...
10865	NGOs	#entity/agents #private ...	PakistanEarthquakeRelief...	10/30/2012 12:49 PM	151	60 The inherent goodnes...
20291	NGOs	#entity/agents #entity/or...	CA in East Africa.txt.xml	11/5/2012 12:55 PM	141	This is true particularly in ...
14587	NGOs	#entity/organizations #pri...	Iraq Building Civil-Military ...	10/30/2012 12:59 PM	131	US forces also aligned th...
14675	NGOs	#entity/organizations #pri...	Interagency Response to ...	10/30/2012 12:59 PM	38	b. Ensure that PRT units ...

Figure 12: Results of Corpus Database Search of Phrase “NGO’s”

Analysis of the Annotations

L1 Tag	Count
#entity/physical_behaviors	0
#entity/physical_infrastructures	396
#entity/social_infrastructures	475
#entity/technical_capabilities	589
#entity/organizations	710
#entity/social_behaviors	725
#entity/institutions	778
#entity/time	842
#entity/events	890
#entity/materials	899
#entity/agents/us/m/ca	1205
#entity/services	1411
#entity/places	2207
#entity/info	2720
#entity/agents/us/m	3144
#entity/agents/us	3757
#entity/agents	4198
Mean	1467
Median	895
STDEV	1260

Figure 3a. Frequency Distribution of L1 Tags

L2 Tag		Count
#return		13
#negotiation		38
#routine		38
#definition		41
#illicit		47
#food		61
#private		74
#requirement		75
#agreement		78
#criminal		90
#public_opinion		98
#contractor		102
#utilities		122
#psychological		142
#global		229
#transportation		235
#agriculture		266
#purpose		296
#religion		307
#political		338
#health		365
#economy		483
#environment		491
#social		492
#extremist		499
#public		553
#education		575
#administrative		577
#condition		596
#perspective		623
#oversight		663
#security		761
#communication		790
#relief		838
#governance		903
#conflicting		1405
#civilian		1622
#authority		1821
#military		3284
#relationship		4140
Mean	604	604
Median	351.5	352
STDEV	843.932	844

Figure 3b. Frequency Distribution of L2 Tags

Figures 3a and 3b provide an insight into document corpus, with the former providing a global view of the document set, and the latter a closer view of the socio-cultural content. For the L1 tags, the median number of occurrences is 895. As expected the standard deviation is large, which indicates the labels are disparate. The tags that occur at a higher rate than the median are as to be expected: *agents*, *agents/us*, *agents/us/m*, *agents/us/m/ca*, *info*, *materials*, *events*, *places*, *services*. *time* is close to the median. The fact that tags such as *agents*, *agents/us*, *agents/us/m*, *agents/us/m/ca* would occur at a higher rate are to be expected, since the documents primarily relate to US Army in operational settings. The other tags in this list *info*, *materials*, *events*, *places*, *services* taken in totality provide broad indications that the document set is focused on supporting operations. The tags with frequency less than the median such as *institutions*, *social_behaviors*, *organizations*, *technical_capabilities*, and *social_infrastructures* further buttress this conclusion, and at the same time indicating that there were documents relating to the social infrastructure, with presumably operations linked to it. *physical_behaviors* was not used as an annotation item- while this may be somewhat surprising, this could result from the fact that *physical_behaviors* may be indicated through synonymous L2 tags. It might also be concluded that none of the documents contained descriptions of the properties of systems, which might require annotation using the *physical_behaviors*.

The L2 tags have a median of 352 and a standard deviation of 844. As in the case of the L1 tags, it is clear that the L2s are independent. The tags that have a frequency higher than the median mostly indicate the military nature of the document set. Tags such as *#military*, *#conficting*, *#authority*, *#relationship*, *#security*, *#extremist* etc., dominate the annotations. However, there are fairly frequent representations of socio-cultural artifacts such as *#political*, *#transportation*, *#agriculture*, *#religion*, *#health*, etc. This would indicate that the document set analyzed contains material pertaining to the non-military mission of the US Army. Analyzing the L2 tags alone, there is fairly good agreement with the broad thrusts seen in the L1 annotations.

In summary, the frequency lists indicate that the document set annotated contains a broad range of topics- there are no outstanding themes that are indicated apart from the Army “relatedness” of the set. This is consistent with the document set, which has a wide range of themes and topics, from agriculture to combat to education, schools, etc. Annotation can provide deeper insight into a document set, however for this process to be more effective, the document set has to be focused rather than generalized. An

approach to providing this focus would be to automatically cluster the document set, and examine the variance of the annotations for each document from the centroid (or the average) of the set.

Lessons Learned

Manual annotation of text, even with software tools is a difficult process. It was difficult to keep the annotators engaged for a length of time. This could be ascribed to the fact that skilled annotators are a particular type of individual, very focused and detail oriented. Quality assurance of the annotations was therefore a challenge, and in this project two separate annotators were used for every document- one annotator, and the second for quality control. Productivity of the annotators was therefore a major challenge. A minor issue arose from the errors introduced during the document pre-processing stage. There were numerous non-standard ASCII characters introduced during this process, requiring multiple iterations over the documents.

The experience with manual annotation has led the CAU team to propose an *incremental approach* to annotation. In the *Incremental Annotation* approach, documents are first annotated to the extent possible. A core document is first annotated, and all subsequent document annotations are compared with the core document. Care is taken that documents in different categories are separately annotated. This would help to maintain quality and uniformity over the annotation by multiple annotators and a large set of documents.

Phase 2: Automating the Annotation Process

Manual annotation is a human intensive process and is not feasible for a large corpus of text. Classification is a technique, well-researched in data mining and machine learning that may be used to automate the annotation process. Classification separates data into distinct classes characterized by some distinguishing features and rules relate class labels to these features. Automated classification is dominant in a variety of domains: text data such as e-mails, web pages, news articles; audio; images and video; medical data; or even annotated genes (Read, 2010). Each example is associated with an attribute vector which represents data from its domain. Labels represent concepts from the problem domain such as subject categories, descriptive tags, genres, gene functions, and other forms of annotation. The training set is readily available in practical scenarios, usually in the form of human-annotation by a domain expert. A supervised classifier trains its model on these examples and continues the labeling task thereafter automatically. Single-label classification is the task of associating each example with a single class label. The classifier learns to associate each new test example with one of the known class labels. Classes may also overlap, in which case, the same data may belong to all of the many classes that overlap. In such instances, it becomes necessary to collect the details or features of all the classes that the data belongs to in order to perform a complete classification that is also accurate. When each example may be associated with multiple labels simultaneously, this is known as multi-label classification. For example, a news article about a conference on renewable energy sources, can be intuitively labeled both science and environment. In this effort, the terms class and label are equivalent and will be used interchangeably.

Different approaches are used to deal with multi-label problems. Some methods transform a multi-label classification problem into a set of single-label classification problems by problem transformation techniques, while using traditional classification algorithms. Other methods develop new algorithms or enhance and adapt specific classification algorithms using algorithm adaptation techniques in order to accomplish the task of multi-label classification.

This report presents the Machine Learning-Based Text Annotation Workbench (MaLTAW), an annotation assistance tool that reduces the difficulty of the annotation process. The area of application for the tool is a corpus supplied by the U.S. Army Corps of Engineers with the objective of annotating the text using a classification taxonomy provided. The corpus consists of numerous reports, lessons learnt and best practices drawn from peace keeping and nation building operations. There are several technical challenges posed by this domain. The document set is complex with respect to size due to the variety of formats and range of subject matter. The subject matter in these documents is extensive and includes social and cultural institutions, infrastructure, education, agriculture, etc. The taxonomy is large and unstructured with the flexibility of labels being applied orthogonally. Consequently, the search space for the label(s) become prohibitively large and becomes necessary to adopt a selection strategy that reduces the complexity of the classification process.

Fully automated annotation of text is a goal that is problematic primarily arising out of the context sensitive nature of text. A practical approach is to develop systems that can assist the manual annotation

process keeping the human in the loop. Additional complexity is introduced by the domain of application as outlined previously. We develop an innovative system, MaLTAW, which uses the Naïve Bayes machine learning as an assistant to the manual annotation of the corpus. We introduce a simplification technique to reduce the massive search space of labels introduced by the domain. We improve precision by supplementing these predictive algorithms with similarity based measures and evaluate MaLTAW for performance using both prediction-based metrics and ranking-based metrics. The performance of MaLTAW is compared and benchmarked against a standard text classification algorithm, the Multi-label k Nearest Neighbor (MLkNN). It is shown that MaLTAW performs better than MLkNN on all evaluation metrics.

This section is organized as follow: the published literature on text processing and multi-label classification in text is reviewed; the approach to the problem and the architecture of MaLTAW is then described; finally the results obtained are compared with alternate approaches.

Related work

We describe previous efforts in the areas of text annotation, classification, and annotation tools, since each of these areas are relevant to this research. (Teufel et al, 1999) use text annotation to clarify the argumentative role of each sentence in the document to develop an automatic text summarization. The annotation scheme focuses on annotating research articles. (Cardie et al., 2008) describe the application of text annotation in political science research. They emphasize the issue of agreement between manual annotation and supervised annotation using learning algorithms. As humans make mistakes, the classifier is also expected to produce less than 100% agreement. As the number of categories or labels increases, percentage agreement or classification accuracy is expected to decline.

In document classification typically a large number of attributes are used. The attributes of the examples to be classified are the words in the text phrases, and the number of different words can be quite large. (McCallum, Nigam, 1998) clarify the two different first order probabilistic generative models that are used for text classification, both of which make the Naïve Bayes assumption. The first model is a multi-variate Bernoulli model which is a Bayesian network with no dependencies between words and binary features. The second model is the multinomial model which specifies that a document is represented by the set of word occurrences in the document. The probability of a document is a product of the probability of each of the words that occur. Individual word occurrences are events and the document is a collection of word events. The multinomial model performs better with larger vocabulary sizes. Several learning algorithms that have been applied to text document classification including Multi-label k-nearest neighbor (MLkNN), Support Vector Machines (SVM), Naive Bayes (NB) etc. All these techniques perform comparably well. MLkNN has the distinguishing characteristic that the algorithm is iterative. SVMs use discriminative techniques and are based on statistical learning. Their training time is quadratic to the number of training examples but they are known to be the most accurate (Godbole, Sarawagi, 2004). Naïve Bayes classifiers are faster as they learn a probabilistic generative model in just one pass of the training data even though they may sacrifice some classification accuracy.

(Lauser et al., 2003) propose an approach to automatically subject index full-text documents with multiple labels based on binary Support Vector Machines (SVMs). The authors incorporate multilingual background knowledge in the form of thesauri and ontologies in their text document representation.

(Godbole, Sarawagi, 2004) present methods for enhancing and adapting discriminative classifiers for multi-labeled predictions. Their approach exploits the relationship between classes, by combining text features and the features indicating relationship between classes. They also propose enhancements to the margin of SVMs for building better models in the event of overlapping classes. In (Goncalves, Quaresma, 2005), the authors evaluate which preprocessing combination of feature reduction, feature subset selection, and term weighting is best suited to yield a document representation that optimizes the SVM classification of particular datasets. (Ikonomakis et al., 2005) describe the text classification process. They describe the vector representation of documents, feature selection, and provide some definitions of evaluation metrics. In (Bao et al., 2007), WordNet ® is used to measure similarity of labels that indicates the semantic similarity between documents. Documents are clustered based on rules into similar groups. (Tsoumakas et al., 2007) give a good introduction to multi-label classification using methods such as algorithm adaptation and problem transformation. The different techniques are compared and evaluated using metrics, after they are applied to classify some benchmarked data sets. (Zhang et al., 2007) present a multi-label lazy learning approach named MLkNN, which is derived from the traditional k-Nearest Neighbor (kNN) algorithm. Using experiments on three different multi-label learning problems, i.e. Yeast gene functional analysis, natural scene classification and automatic web page categorization, the authors show that MLkNN achieves better performance when compared to some well-established multi-label learning algorithms. (Carvalho et al., 2009) present a good tutorial on all the multi-label classification techniques. They describe with examples the problem transformation approach that includes label-based transformation and instance-based transformation, and also the algorithm adaptation approach. (Zhang et al., 2009) address the multi-label problem by using a method called MLNB (Multi Label Naïve Bayes) which adapts the traditional naïve Bayes classifier to deal with multi-label instances. Feature selection mechanisms are incorporated into MLNB to improve its performance. Experiments on synthetic and real-world data show that MLNB achieves comparable performance to other well-established multi-label learning algorithms. (Cerri et al., 2009) describe the application of multi-label classification in bioinformatics. Protein function classification is a typical example of multi-label classification, as a protein can have more than one function at a time. (Chang et al., 2011) propose a tree decomposition approach for solving large scale multi-label classification problems. The problem is transformed into a number of “one against others” classification problems. In order to solve each of the smaller problems, a decision tree is used to decompose the corresponding data space and train local SVMs on the decomposed regions. (Younes et al., 2011) describe an adaptation of MLkNN that takes into account dependencies between labels (DMLkNN). The authors use a Bayesian version of kNN. Experiments on simulated and benchmarked datasets show the efficiency of this approach compared to other existing approaches. (Tsoumakas et al., 2011) describe a new enhancement on the multi-label algorithm called label powerset (LP) that considers each distinct combination of labels that exist in the training set as a different class value in a single-label classification task. When the number of classes becomes large and many classes are associated with very few training examples, the initial set of labels is broken into a number of small random subsets called labelsets and LP is used to train corresponding classifiers. The labelsets could be disjoint or have overlap. They propose a method called RAKEL (Random k Label Sets) where k is the parameter that specifies the size of the subsets. RAKEL compares well with other methods.

There has been some previous work in work benches for text annotation. (Koivunen, 2005) describes Annotea, a semantic web-based project. Metadata is generated in the form of objects such as web annotations, reply threads, bookmarks, topics etc. As a result, users can easily create RDF metadata that may be queried, merged and mixed with other metadata. In (Zeni et al., 2007), a software tool (Biblio) is described for automatically generating a list of references and an annotated bibliography, given a collection of published research articles. (Finlayson, 2011) describes the Story Workbench, a software tool that facilitates semantic annotation of text documents. The tool uses Natural Language Processing tools to make a best guess as to the annotation, presenting that guess to the human annotator for approval, correction, or elaboration. This is a semi-automatic process. Annotation is generalized into a “tagging” procedure with parts-of-speech tags as well as general tags for “tooltips” or “infotips” in a GUI.

The problem that we address in this research is unique to the domain in two respects- the need to annotate at an atomic level, i.e., the noun phrase and verb phrase level, and the unstructured labeling taxonomy supplied to annotate text. The taxonomy gives rise to a very large labeling search space, which makes accurate classification of text infeasible. The software tools and algorithms discussed in literature cannot adequately handle these problems.

Approach

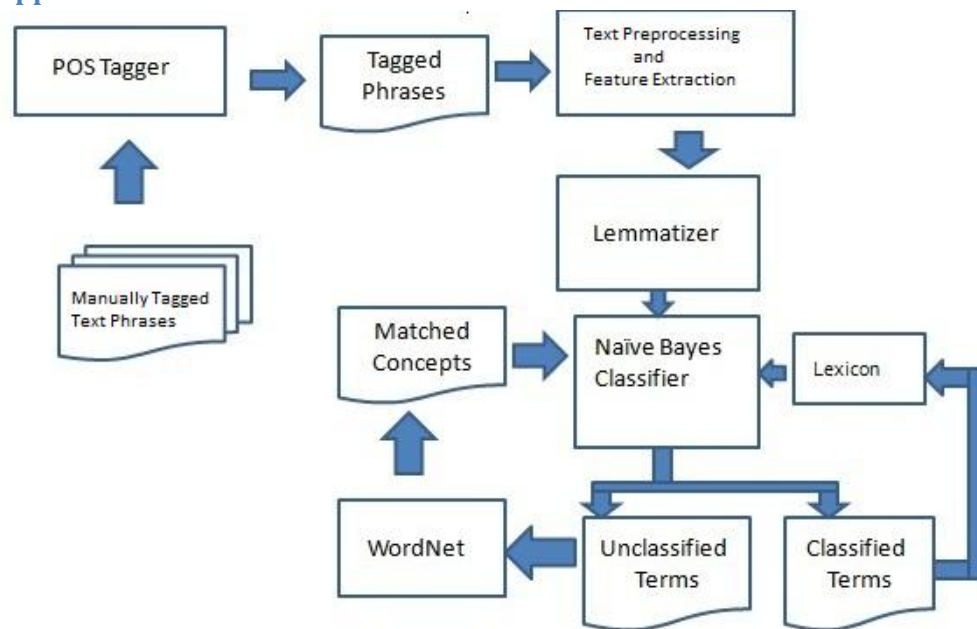


Figure 13. Process Flow Diagram for MaLTAW

Figure 13 shows the process flow diagram for MaLTAW. The modules in Figure 5 including the Naïve Bayes Classifier are implemented in Java®. Manually tagged text phrases are input to the system with Verb phrases and Noun phrases tagged with appropriate Verb Phrase (VP) tags and Noun Phrase (NP) tags respectively. These text phrases are input to the Parts Of Speech (POS) Tagger in Figure 1. The Stanford NLP POS Tagger (POSTagger, 2012) is used for POS Tagging the input phrases. The Text Preprocessor and Feature Extraction component performs pre-processing on POS Tagged data. Pre-processing includes steps such as the filtering of records that do not contain either noun phrases or verb

phrases, and retaining only those features (words) that have appropriate parts-of-speech tags for noun phrases and verb phrases. This component produces as output a delimited ASCII text file for next phase of lemmatizing. The lemmatizer uses WordNet® (WordNet, 2012) database to extract synonyms or lemmas of input phrases to build an expanded input set for next stage of classification. The Java API for WordNet Searching (JAWS, 2012) interface to WordNet is utilized in the lemmatizer. The lemmatized phrases are input into the Naïve Bayes classifier. The lexicon in Figure 1 is a SQLite database (SQLite, 2012) that stores the training data. Unclassified text is passed into WordNet to extract synonyms or matched concepts and returned to the Naïve Bayes Classifier for another attempt at classification.

Naïve Bayes is a standard algorithm for learning to classify text. Naïve Bayes classifiers are faster than other algorithms discussed in literature such as SVMs, since they learn a probabilistic generative model in just one pass of the training data even though they may sacrifice some classification accuracy. The algorithm determines probability of outcome (class) based on conditional probability using the Bayes theorem. Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. If B represents the dependent event and A represents the prior event, Bayes' theorem can be stated as follows in equation (1).

$$\text{Prob (B given A)} = \text{Prob (A and B)}/\text{Prob (A)} \quad (1)$$

To calculate the probability of B given A, the algorithm counts the number of instances where A and B occur together and divides it by the total number of instances in which A occurs. B in this context represents a text phrase while A represents the label corresponding to this text phrase. The Training data consists of text phrases classified /annotated by human annotator which is input into the MaLTAW system as discussed previously.

Classification is be formulated as Single-label (Multi-class) classification or Multi-label classification as described below, where D is the data set and L is the set of labels.

Single-label Classification:

Example phrases are $D=\{x_1, \dots, x_n\}$

Classification Labels are $L = \{l_1, \dots, l_m\}$

Each example is associated with one label: $(x, l \in L)$

Multi-label Classification:

Examples phrases are $D=\{x_1, \dots, x_n\}$

Labels are $L = \{l_1, \dots, l_m\}$

Each example is associated with a subset of labels, $S: (x, S \subseteq L)$

Several problem transformation techniques are described in (Tsoumakas et al., 2007) and (Carvalho et al., 2009). We use the method, PT3, (Tsoumakas et al., 2007) which employs the conjunction of multiple labels to implement an instance-based transformation. The purpose of the transformation is to reduce the number of labels. For example, consider the following data. We have four examples or documents that belong to one or more classes in a set of four classes: Science, Environment, Politics, and Sports.

Example 1 = {Science, Environment}

Example 2 = {Politics, Sports}

Example 3 = {Environment, Politics}

Example 4 = {Sports}

The above set of examples is transformed using the conjunction operator into examples below.

Example 1 = { Science[^] Environment}

Example 2 = {Politics[^]Sports}

Example 3 = {Environment[^]Politics}

Example 4 = {Sports}

This transforms the multi-label problem into a single-label classification problem. A single-label classification algorithm such as Naïve Bayes can now be applied to classify the examples. A disadvantage of PT3 is that it may lead to data sets with a large number of classes and only a few examples per class.

The labels or tags in this research are classified into verb phrase tags (task, state, role, and other) and noun phrase tags. Noun phrase tags are further subdivided into Level 1 (L1), and Level 2 (L2) tags shown in Table 1 and Table2 respectively. We have a very large combination of labels which could be used to tag noun phrases, considering that L1 and L2 tags may be orthogonally applied multiple number of times (depending on the context that we are annotating). The only constraint for applying the tags is that every NP has to have at least one L1 tag. The classification component tries to learn the appropriate L1L2 combination for a phrase based on previously classified phrases. There are $(2^{63} - 1)$ possible combinations of L1 and L2 tags for a given phrase, and the classifier uses previously classified phrases to determine what the appropriate tag(s) might be. The learning of the right annotation is search problem- we search the space of possible solutions to identify the best solution. This search space is very large, and populated sparsely, i.e., the vast majority of the possible labels (annotations) do not have an exemplar phrase. Learning this entire solution space is infeasible, and any learning attempted would result in poor classification accuracy.

Frequency counts (in percentages) are shown for tags in Table 4 and Table 5.

Class Tag	Frequency Counts(%)
entity/info	28.1
entity/agents	23.9
entity/services	10.6
entity/materials	8.6
entity/events	8.6
entity/places	8.1
entity/organizations	6.3
entity/institutions	2.5
entity/time	2.1
entity/technical_capabilities	1.2

Table 4. L1 Tags Frequency Counts.

Class Tag	Frequency Counts (%)
Task	36.5
State	35.3
Other	23.5
Role	4.7

Table 5. VP Tags Frequency Counts.

We employ a simplification strategy to overcome this problem. Rather than learn the entire search space, we break the break it up into component spaces, and learn a subset of the component spaces, i.e., the objective is to discover and learn the more populated regions of the search space. However, it should be noted that while this is a practical strategy, it can reduce recall (see Table 5). To compensate for this multiple learning models are used, each of which focuses on a particular sub-space of the search space, to annotate the text phrases. Also, not that there is no unique way of constructing the model, it could be mechanistic or based on domain knowledge. This is a reasonable strategy, since the annotator is final authority that decides the annotation, though the system provides suggestions.

In this work we employed a strategy that is mechanistic (as opposed to derived from domain knowledge) and focused on widely used annotations. For example, in the learning model for L1 labels Table 7 is used to identify the top ten L1 labels. Using Table 7 as guidance, the previously annotated phrase-annotation pair is modified to a reduced annotation.

In the following hypothetical example,

Phrase: “local building materials”

L1 Manual Annotation: {#entity/materials #entity/agent/physical_infrastructure}

L1 Learning Model Annotation: {#entity/materials}

A more complex model is used for L2 labels. The learning models constituted by L2 labels alone produced poor classification results. Instead we constructed a set of L1L2 models- by similarly reducing the L2 annotations to most frequently. For instance an L2 label {#civilian #relief} would be transformed to {#civilian}, while reducing the L1 label sets as previously described. The objective in these models is to minimize the learning complexity by reducing the number of labels, with each model focusing on the correctness of classification with respect to the particular set of annotations alone.

MaLTAW provides an infrastructure where these models may be used either individually or in combination. The outputs of the models are composed together in the suggested tag pane. The execution of a model is rapid, so it would be possible for a domain expert to construct few narrowly focused models or alternatively large numbers of more generalized models to annotate a targeted text corpus.

The next section presents results of classification and evaluation metrics that evaluate the performance of MaLTAW with different data sets from our DoD application. Comparison of performance is made with MLkNN using the same train-test data.

Results and Validation

MULAN (Tsoumakas et al., 2011) has implemented several algorithms such as MLkNN (Multi-label lazy learning k-NN), RAKEL, HMC, HOMER, Hierarchy Builder, Binary Relevance, Label Power Set etc. The software (written in Java) also generates classification metrics automatically when supplied with train-test data. However, for our real-world data that contains phrase strings derived from WordNet which has an exhaustive vocabulary from A through Z, only MLkNN could be applied successfully. The string valued phrases attribute cannot be converted to other data types such as nominal attributes in Weka (Machine Learning Group, 2012). Also, for our data, the labels do not have any intrinsic structure such as hierarchical structure. The following results therefore only record the metrics obtained from MLkNN algorithm in MULAN.

Evaluation Metrics

Prediction-based metrics and Ranking-based metrics are standard measures used to evaluate performance of text classification. Ranking based metrics (Table 4) evaluate the label ranking quality depending on the ranking or scoring function. Hamming Loss is used as the basis for Ranking Function in our classification. Lower Hamming Loss implies higher rank for a label. The most relevant label has highest rank of 1. Prediction-based metrics assess the correctness of the label sets predicted by the multi-label classifier (Table 5).

1. Subset Accuracy =
(No. of Exact Matched records with True Predicted Classes)/(No. of Test Records)
2. Average Precision: Average fraction of labels ranked above a particular label (Best value is 1)
3. Coverage: Average # of steps needed to move down the ranked label list in order to cover all the labels assigned to a test instance. Smaller value of this metric is desirable.
4. One-Error: This metric calculates how many times the top-ranked label i.e. the label with highest ranking score, is not in the set of labels for the appropriate instance. Smaller value of this metric is desirable.
5. Ranking loss: Average fraction of label pairs that are reversely ordered i.e. number of times irrelevant labels are ranked higher than relevant labels for an instance. This does not happen in our case. Smaller value of this metric is desirable.

Table 6. Definitions of Ranking-based metrics

1. Hamming Loss = $\frac{\text{\# of misclassified records in Test Data}}{(\text{\# of records of Test Data} * \text{Size of Label Set})}$
2. Label based Accuracy = $\frac{\text{\# of correctly classified records in test data}}{((\text{Size of Test Data}) * \text{Maximum}(\text{Size of Predicted labels Set}, \text{Size of True labels Set}))}$
3. Label Cardinality = $\frac{\text{Sum of all labels applicable to each record of data for the test records}}{(\text{\# of records in the Test Data})}$
4. Label Density = $\frac{\text{Label Cardinality}}{(\text{Size of True Labels Set})}$
5. DL(D) = # of Distinct Label Sets¹
6. Percentage Classification Accuracy = $\frac{\text{\# of correctly classified records in Test Data} * 100}{(\text{Total number of records in Test Data})}$
7. Macro Measures (Label-based)
 - a) Precision = $\frac{1}{(\text{size of Predicted Labels' Set}) * \Sigma (\text{\# of correctly classified records in each class}) / (\text{\# of Predicted records})}$
 - b) Recall = $\frac{1}{(\text{size of True Labels' Set}) * \Sigma (\text{\# of correctly classified records in each class}) / (\text{\# of Actual records})}$
 - c) F1 Score = $\frac{(1 / \text{Size of True Labels' Set}) * \Sigma (2 * \text{Precision}(y) * \text{Recall}(y))}{(\text{Precision}(y) + \text{Recall}(y))}$ Micro Measures
 - a) Precision = $\frac{1}{(\text{\# of records in test data}) * \Sigma (\text{\# of correctly classified records in Test Data}) / (\text{\# of Predicted Classes})}$
 - b) Recall = $\frac{1}{(\text{\# of records in test data}) * \Sigma (\text{\# of correctly classified records in Test Data}) / (\text{\# of Actual Classes})}$
 - c) F1 Score = $\frac{(1 / \text{\# of records in test data}) * (2 * \text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$

Table 7. Definitions of Prediction-based metrics

¹# of Distinct Label Sets is 1 in our tests as full set of level 1 and full set of level 2 Tags are not considered. Only top 10 classes are considered in NP whether level 1 or level 2 most frequent, second most frequent, third most frequent. Level 1 and Level 2 Tags are combined. All 4 classes of VP tags are included in the tests.

²y is a label or class in the True Labels Set and Summation is calculated over all labels in the True Labels Set

Testing the Approach

One L1 and four L1L2 models are generated to test the simplification approach described in the previous section. They are generated using the frequency of the L2 tag in the entire manually annotated set. For example the L1L2⁽¹⁾ set in conjunction with the L1 tag, has the most frequent L2 tag alone, the L1L2⁽²⁾ set has the second most frequent L2 tag, and so on. (The superscript notation is used to indicate the frequency of the L2 tag). The different sets of L1L2 tags are generated and used to test the validity of this approach. Table 8 describes the experimental set up for the system detailing the number of training and test instances.

Type of Classification	Data Set	Train instances	Test instances	Number of classes
NP	L1	2096	434	10
	L1L2 ⁽¹⁾	1002	234	16
	L1L2 ⁽²⁾	1658	364	13
	L1L2 ⁽³⁾	2068	413	12
	L1L2 ⁽⁴⁾	1241	315	9
VP	VP	833	200	4

Table 8. Description of data sets used in the tests

Prediction-based metrics

In the first part of the experimental set up, we consider L1 Noun Phrase (NP) tags alone and in a separate experiment Verb Phrases alone. The results of this experiment are shown in Table 13.

Evaluation Metric	L1	VP
Classification Accuracy (%)	70	80
Hamming Loss	0.03	0.05
Label-based Accuracy	0.07	0.2
Label Cardinality	2.088	2.17
Label Density	0.2088	0.543
Macro Precision	0.9156	0.858
Macro Recall	0.3652	0.386
Macro F1 Score	0.4878	0.529
Micro Precision	0.07	0.2
Micro Recall	0.07	0.2
Micro F1 Score	0.00016	0.001

Table 9. Evaluation Metrics from Naïve Bayes using only L1 Tags for Noun Phrases and VP Tags for Verb Phrases.

Prediction Metrics	L1L2 ⁽¹⁾	L1L2 ⁽²⁾	L1L2 ⁽³⁾	L1L2 ⁽⁴⁾
Classification Accuracy or Subset Accuracy (%)	58.55	64.84	71.7	54.92
Hamming Loss	0.026	0.027	0.0236	0.05
Label-based Accuracy	0.037	0.05	0.0597	0.061
Label Cardinality	1.962	2.17	2.242	1.654
Label Density	0.123	0.167	0.187	0.184
Macro Precision	0.943	0.9175	0.93	0.906
Macro Recall	0.288	0.264	0.3156	0.37
Macro F1 Score	0.404	0.378	0.4456	0.365
Micro Precision	0.037	0.054	0.0597	0.061
Micro Recall	0.037	0.05	0.0597	0.061
Micro F1 Score	0.000158	0.000143	0.000145	0.000194

Table 10. Prediction Metrics in Naïve Bayes using the L1L2⁽ⁱ⁾ set.

For the Naïve Bayes classifier, with NP (Noun Phrase) data, with a threshold applied, 106 records remained unclassified and 128 records (spanning 15 classes) were accurately classified for the data set with L2⁽¹⁾ tags combined with L1 tags. Size of test data used in all metrics calculations is reduced to 128 records for this data set. For the data set with L2⁽²⁾ tags combined with L1 tags, the threshold was set and consequently, 129 records remained unclassified and 235 records (spanning 11 classes) were accurately classified. Size of test data used in all metrics calculations is reduced to 235 records for this data set. For the data set with L2⁽³⁾ tags combined with L1 tags, the threshold was set and consequently, 135 records remained unclassified and 278 records (spanning 11 classes) were accurately classified. Size of test data used in all metrics calculations is reduced to 278 records for this data set. For the data set with L2⁽⁴⁾ tags combined with L1 tags, the threshold was set and consequently, 168 records remained unclassified and 147 records (spanning 8 classes) were accurately classified. Size of test data used in all metrics calculations is reduced to 147 records for this data set. Table 14 summarizes the evaluation metrics calculated in each test case.

From Table 11, it may be noted that those instances that are classified, are classified accurately. The remaining instances remain unclassified. Hamming Loss is zero in all cases. Label-based accuracy is best for the test set with L1L2⁽⁴⁾ tag. Label cardinality is highest for the test set with L1L2⁽²⁾ tag. Label density is highest for the test set with L1L2⁽⁴⁾ tag. Macro Precision is uniformly 1 in all cases. Macro Recall is highest for the test set with L1L2⁽⁴⁾ tag. Macro F1 Score is highest for the test set with L1L2⁽⁴⁾ tag. Micro Precision, Micro Recall and Micro F1 Score are all highest for the test set with L1L2⁽⁴⁾ tag.

In each test data set (each column in Table 15), as the threshold is applied in the NB classifier, the class that has maximum number of mismatches (all instances for the class considered), is eliminated. This results in the increased classification accuracy shown in the table.

Evaluation Metric	L1L2⁽¹⁾	L1L2⁽²⁾	L1L2⁽³⁾	L1L2⁽⁴⁾
Classification Accuracy (%)	100	100	100	100
Hamming Loss	0	0	0	0
Label-based Accuracy	0.0625	0.077	0.083	0.11
Label Cardinality	2.62	2.78	2.745	2.37
Label Density	0.164	0.214	0.2288	0.263
Macro Precision	1	1	1	1
Macro Recall	0.3657	0.3218	0.367	0.436
Macro F1 Score	0.495	0.434	0.4972	0.56
Micro Precision	0.067	0.091	0.091	0.125
Micro Recall	0.0625	0.077	0.083	0.11
Micro F1 Score	0.000505	0.000355	0.0003123	0.0008

Table 11. Prediction Metrics from Naïve Bayes using the L1L2⁽ⁱ⁾ set.

Ranking-based metrics

Classification Algorithm	Ranking Metrics	L1 Tags only for Noun Phrases	Verb Phrase Tags only
Naïve Bayes	Hamming Loss	0.03	0.05
	Subset Accuracy	0.7	0.8
	Average Precision	0.78	0.5
	Coverage	2	1
	One Error	0.0023	0.005
	Ranking Loss	0	0
MULAN MLkNN	Hamming Loss	0.1	0.25
	Subset Accuracy	< 0.01	< 0.01
	Avg. Precision	0.53	0.5454
	Coverage	2.1336	1.255
	One Error	0.6889	0.76
	Ranking Loss	0.2371	0.4183

Table 12. Ranking Metrics and using L1 Noun Phrases and Verb Phrases.

Table 12 shows the evaluation metrics that are ranking based for two classification algorithms: our native Naïve Bayes and MULAN's MLkNN. Overall, the Naïve Bayes algorithm has better performance. MLkNN has marginally better Average Precision for Verb Phase test data. All other evaluation metrics are considerably better for our native Naïve Bayes classifier.

Classification Algorithm	Ranking Metrics	L1L2⁽¹⁾	L1L2⁽²⁾	L1L2⁽³⁾	L1L2⁽⁴⁾
Naïve Bayes	Hamming Loss	0.026	0.027	0.0236	0.05
	Subset Accuracy	0.59	0.65	0.72	0.55
	Average Precision	0.91	0.92	0.75	0.89
	Coverage	4	3	3	2
	One Error	0.0043	0.0055	0.0024	0
	Ranking Loss	0	0	0	0
MULAN MLKNN	Hamming Loss	0.0625	0.0769	0.1	0.1111
	Subset Accuracy	< 0.01	<0.01	<0.01	<0.01
	Average Precision	0.4448	0.4863	0.4883	0.5208
	Coverage	4.2735	3.1209	2.4633	2.2038
	One Error	0.6795	0.7005	0.7267	0.6943
	Ranking Loss	0.2849	0.2601	0.2737	0.2755

Table 13. Ranking Metrics in Naïve Bayes using the L1L2⁽ⁱ⁾ set.

Table 13 shows the evaluation metrics that are ranking based for two classification algorithms: the native Naïve Bayes and MULAN's MLkNN. In all cases the Bayes algorithm has better performance. The test data consists of noun phrases with combined L2 and L1 tags. It is noted that subset accuracy for MULAN MLkNN is less than 0.01 for our data.

The automation process in this domain is complicated since the annotation taxonomy is unstructured. However, it may be concluded that the Naïve Bayes approach provides a good approach to the problem of automated annotation of text. We have validated the algorithmic approach as well as the simplification methodology through extensive testing, and comparison with a benchmark algorithm. It should be noted

that it is difficult to make generalized conclusions about this approach across taxonomies and different text collections. Any inferences on the feasibility of the approach are with respect to the annotation taxonomy provided and the text corpus the taxonomy was applied to.

Integrating the Learning Component into the Workbench

The previous section detailed the development of the machine learning component for the text corpus, its testing and validation. This learning component is integrated into the software tool, to develop the Machine Learning for Text Annotation Workbench (MaLTAW). MaLTAW uses the learning component, and the corpus database to provide hints for annotation to the user. The schema for the corpus database is shown in Figure 6. SQLite is chosen as the database for the work bench because of its light weight footprint. The workbench is developed using the .NET framework, with the learning components constructed as Java services. The user interface of the Workbench is shown in Figure 4. Note that the Workbench support both manual and automated annotation. The latter currently is implemented as a system providing hints to the user, aiding them in this process.

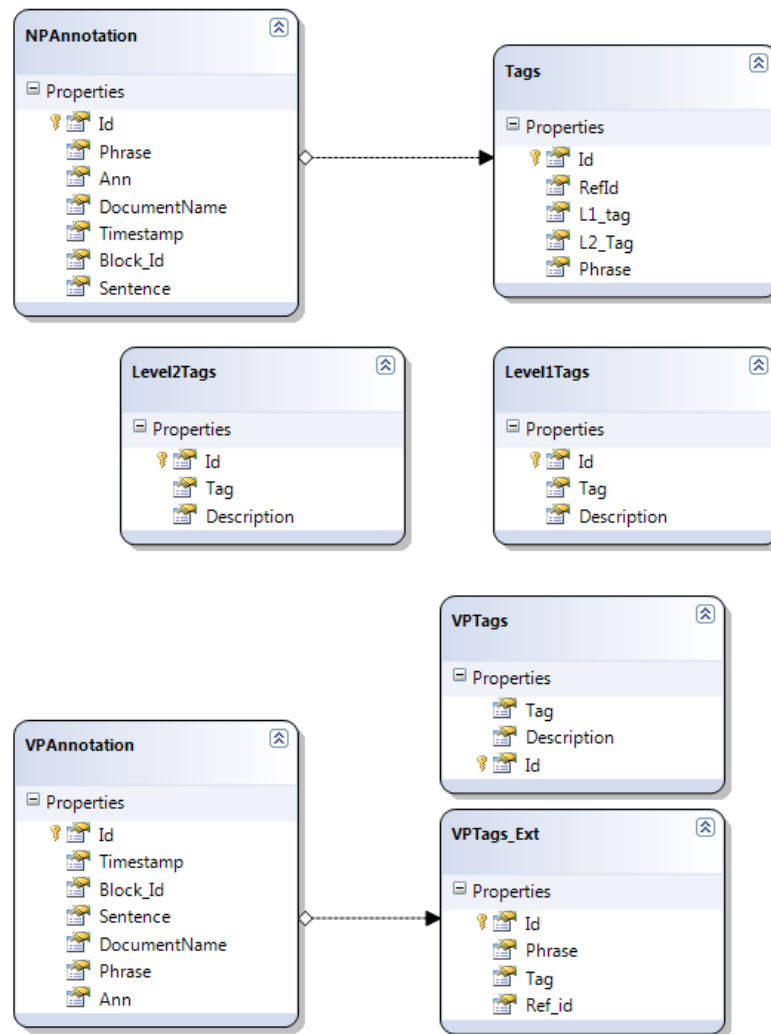


Figure 14: Schema of the Corpus Database

Testing a User-Defined Model

User-defined models may be tested using the **Test Configuration** tab (Figure). The user provides the **Phrase** and **Type** (Noun Phrase/Verb Phrase) information and clicks the **Annotate** button. This option is used to increase the confidence in a particular model and to tweak the model by adding new phrase-annotation pairs. A sample output in the **Test Configuration** mode is shown in Figure

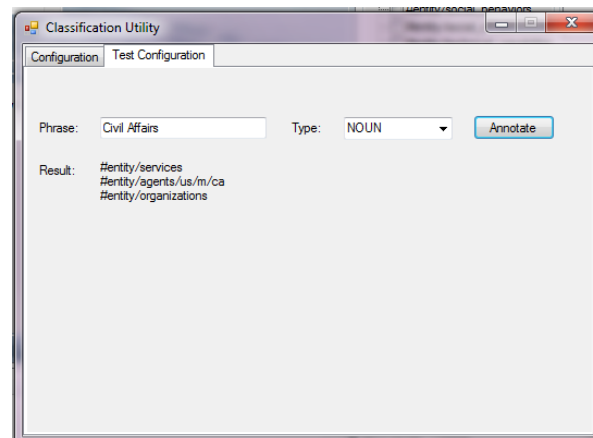


Figure 16: Test Configuration of NP/VP Model

Other Utility Functions

Exporting the Database to a Text File

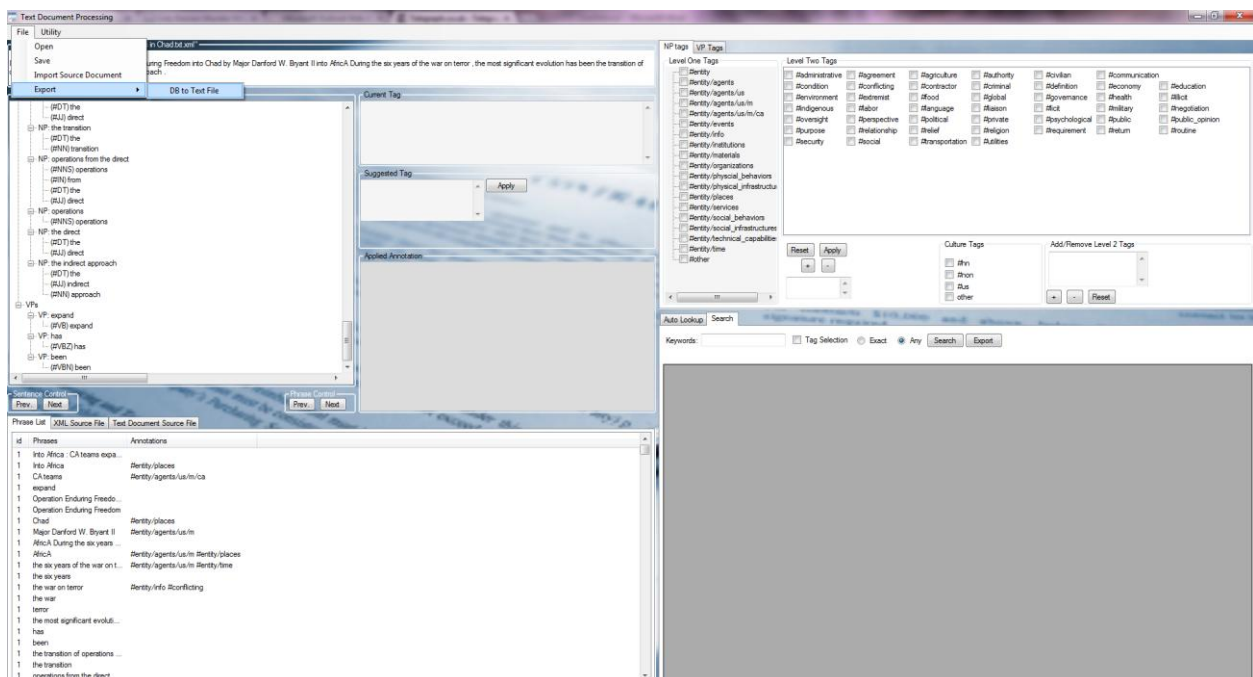
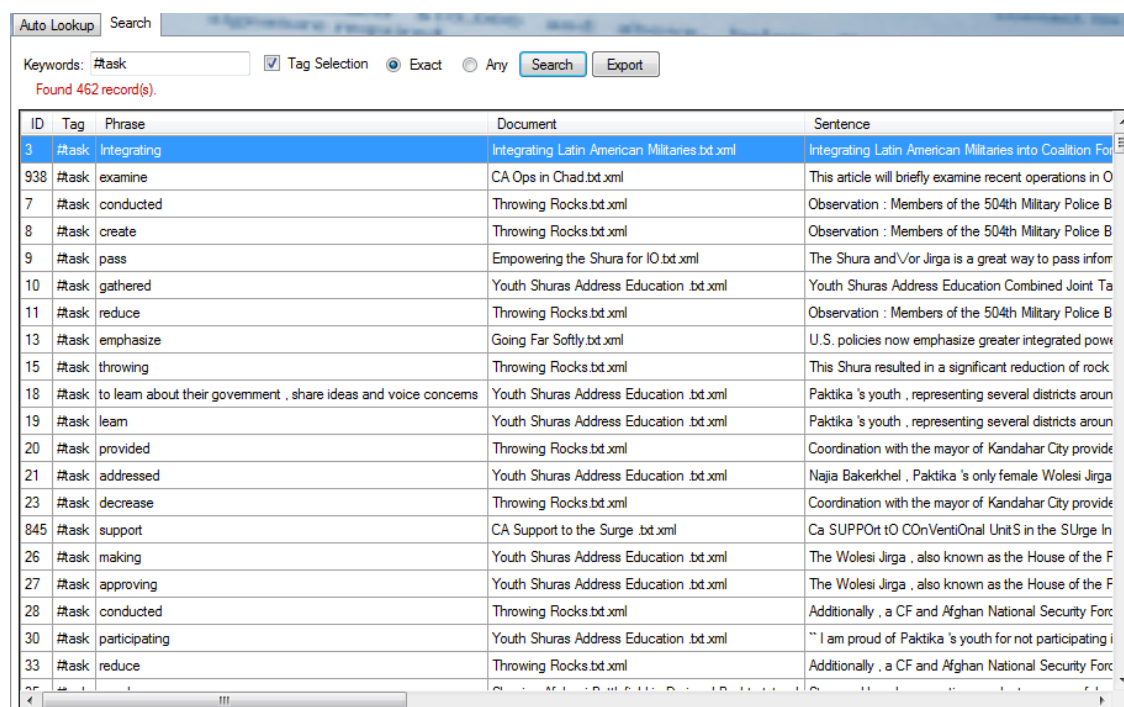


Figure 17: Exporting Corpus Database to a Text File

The **File/Export/DB to Text File** option gives the user the ability to export the database to a Text File. This action may be used to perform additional analytics on the corpus database. An alternate option, with more query control is also available as shown in Figure 15. Here we use the **Search** function in the Corpus Database Pane. This option may be used to retrieve phrases or annotations and the resulting output may be exported as a text file. Figure 15 shows a sample query on the Corpus Database for the annotation tag #task.



ID	Tag	Phrase	Document	Sentence
3	#task	Integrating	Integrating Latin American Militaries.txt.xml	Integrating Latin American Militaries into Coalition Forces
938	#task	examine	CA Ops in Chad.txt.xml	This article will briefly examine recent operations in O
7	#task	conducted	Throwing Rocks.txt.xml	Observation : Members of the 504th Military Police B
8	#task	create	Throwing Rocks.txt.xml	Observation : Members of the 504th Military Police B
9	#task	pass	Empowering the Shura for IO.txt.xml	The Shura and/or Jirga is a great way to pass inform
10	#task	gathered	Youth Shuras Address Education .txt.xml	Youth Shuras Address Education Combined Joint Ta
11	#task	reduce	Throwing Rocks.txt.xml	Observation : Members of the 504th Military Police B
13	#task	emphasize	Going Far Softly.txt.xml	U.S. policies now emphasize greater integrated powe
15	#task	throwing	Throwing Rocks.txt.xml	This Shura resulted in a significant reduction of rock
18	#task	to learn about their government , share ideas and voice concerns	Youth Shuras Address Education .txt.xml	Paktika 's youth , representing several districts aroun
19	#task	learn	Youth Shuras Address Education .txt.xml	Paktika 's youth , representing several districts aroun
20	#task	provided	Throwing Rocks.txt.xml	Coordination with the mayor of Kandahar City provide
21	#task	addressed	Youth Shuras Address Education .txt.xml	Najia Bakerkhel , Paktika 's only female Wolesi Jirga
23	#task	decrease	Throwing Rocks.txt.xml	Coordination with the mayor of Kandahar City provide
845	#task	support	CA Support to the Surge .txt.xml	Ca SUPPOrt to ConVentional Units in the Surge In
26	#task	making	Youth Shuras Address Education .txt.xml	The Wolesi Jirga , also known as the House of the F
27	#task	approving	Youth Shuras Address Education .txt.xml	The Wolesi Jirga , also known as the House of the F
28	#task	conducted	Throwing Rocks.txt.xml	Additionally , a CF and Afghan National Security Forc
30	#task	participating	Youth Shuras Address Education .txt.xml	" I am proud of Paktika 's youth for not participating i
33	#task	reduce	Throwing Rocks.txt.xml	Additionally , a CF and Afghan National Security Forc

Figure 18: Retrieving Phrases by Annotation

Conclusions and Future Work

The US Army is engaged in operations that require an understanding of the spatial, cultural, and social factors that are motivating factors for the participants in such scenarios. The understanding of social and cultural structures and of the relationships between them is vital in population-centric operations. However, much of this information while recorded is locked in textual or other unstructured data formats and inaccessible to decision makers. Computational tools that can extract such information are central to improving decisions at all levels. Decision makers are then empowered to search, evaluate, and act upon the information, which is now amenable to inferencing, and other logical operations. This contract focused on the development of annotation tools that provide the enabling technology required for this purpose.

This work approached the problem of text annotation in two phases. In the first phase we annotated the text manually based on a taxonomy provided, and in the second phase we used the annotations to develop

algorithms to perform automated annotation. There were several challenging components to developing the automated annotation component- the text corpus is wide ranging encompassing a broad range of topics; the taxonomy is unstructured and large; and finally the annotations may be applied combinatorially. We introduced a multi-modal approach that reduces the combinatorial nature of the problem, making the automation feasible. Finally, we combined these techniques into a novel integrated system, Machine Learning for Text Annotation Workbench (MaLTAW) that facilitates both manual annotation and incorporates supervised annotation to reduce the complexity of annotating text. MaLTAW provides a flexible environment with the ability to change the taxonomy depending on the domain of application. The MaLTAW tool also has the capability to ensure a consistent basis to annotation, since it generates annotations based on the deterministic learning component. This infrastructure is therefore ideal to implement the *Incremental Annotation* process introduced earlier.

While the MaLTAW tool provides a good infrastructure for annotation of text, there are several possible enhancements to the tool that can improve the quality and repeatability of the annotation process.

- Automated generation of learning models. The quality of the automated annotations provided by MaLTAW depends on the learning model used (refer the MaLTAW user guide). Currently the learning model is generated manually, however the corpus database provides a mechanism for the automated creation of models. Multiple models may be created and used within MaLTAW. The use of multiple models would improve the recall of the classification process.
- Qualitative evaluation of model outputs. In the current version of MaLTAW every model output is treated identically. A quantitative approach that evaluates each model independently, and ranks the outputs could provide the user with greater confidence in the results. This would also help the user select annotation suggestions based on quantitative measures.
- Traceability of MaLTAW suggestions. In the current version of MaLTAW, the annotation suggestions are provided to the user, however there is no mechanism that informs the user as to how the suggestion was being made. Providing traceability of the model outputs would improve confidence in the system.
- Enhancement of MaLTAW to a Web Service. Having MaLTAW as a Web Service would permit multiple annotators to work simultaneously in a document while availing of the infrastructure features provided by MaLTAW, such as the standardization of annotations.

Other extensions to this work may be considered to improve the quality of automation. Documents could be clustered as a first step before the annotation process. This would categorize documents into sets, which could then be annotated similarly. Intuitively, we feel that the quality of the annotations could be enhanced using this process. Structuring the taxonomy into a hierarchy could also improve the quality of annotation. The hierarchical structure could be then used as a basis for model generation, thereby providing a mechanism for reducing classification complexity. Finally, we may consider the annotation as the first step in a process of understanding text. The annotations themselves may be used as the basis to generating higher order phrases such as *Agent* $\rightarrow$ *Action*. The transformation of low level annotations to these type of template phrases is however complex and will require additional investigation.

References

- (Bao, J.P., Lyon, C.M., Lane, P.C.R., 2007). "A text annotation method based on semantic sequence," in *Proceedings of the Seventh International Workshop on Computational Semantics*.
- (Cardie, C., Wilkerson, J., 2008). *Journal of Information Technology and Politics*. 5(1):1-6.
- (Cerri, R., Da Silva, R.O. R., De Carvalho, A. C.P.L.F., 2009). "Comparing Methods for Multilabel Classification of Proteins using Machine Learning Techniques," *Proceedings of 4th Brazilian Symposium on Bioinformatics* (BSB July 2009).
- (Chang, F., Liu, C.C., 2011). "Solving Large-Scale Multi-Label SVM Problems with A Tree Decomposition Approach," *JMLR: Workshop and Conference, ACML2011*.
- (De Carvalho, A.C.P.L.F., Freitas, A.A., 2009) "A tutorial on multi-label classification techniques . Foundations of Computational Intelligence" of *Studies in Computational Intelligence 205*, pages 177-195 Springer, September 2009.
- (Finlayson, M.A., 2011). The Story Workbench: An Extensible Semi-Automatic Text Annotation Tool. *AAAI Technical Report WS-11-18*. Copyright 2011.
- (Godbole, S., Sarawagi, S., 2004) Discriminative Methods for Multi-labeled Classification. *PAKDD 2004*, LNAI 3056, pp. 22–30.
- (Gonçalves, T., Quaresma, P., 2005). Evaluating preprocessing techniques in a text classification problem. *XXV Congresso da Sociedade Brasileira de Computação* (July 2005), pp. 841-850
- (Ikonomakis, M., Kotsiantis, S., Tampakas, V., 2005). Text Classification: A Recent Overview. ICCOMP'05. *Proceedings of the 9th WSEAS International Conference on Computers*. Article no: 125. In ESWC 2005, UserSWeb workshop
- (JAWS, 2012). <http://lyle.smu.edu/~tspell/jaws/index.html>
- (Koivunen, M.R., 2005). "Annotea and semantic web supported collaboration," *Downloaded from* http://ceur-ws.org/Vol-137/01_koivunen_final.pdf, November 2012.
- (Lauser, B., Hotho, A., Koch, T., Solvberg, I.T. (Eds.), 2003). Automatic Multi-label Subject Indexing in a Multilingual Environment. *ECDL 2003*, LNCS 2769, pp. 140–151.
- (Machine Learning Group at University of Waikato, 2012). <http://www.cs.waikato.ac.nz/ml/weka/>
- (McCallum, A., Nigam, K. A., 1998). Comparison of Event Models for Naive Bayes Text Classification. In *AAAI-98 Workshop on 'Learning for Text Categorization'*
- (POSTagger, 2012). <http://nlp.stanford.edu/software/tagger.shtml>
- (Read, J., 2010). *Scalable Multi-label Classification*. Ph.D Thesis, University of Waikato, Hamilton, New Zealand, September 2010.
- (SQLite, 2012). <http://www.sqlite.org/>

(Teufel, S., Carletta, J., Moens, M., 1999). An annotation scheme for discourse-level argumentation in research articles. Proceedings of EACL '99.

(Tsoumakas, G., Katakis, I., Vlahavas, I., 2011). Random k-Labelsets for Multilabel classification. IEEE Transactions on Knowledge and Data Engineering, Vol 23, No. 7, July 2011.

(Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I., 2011). MULAN: A Java library for Multi-Label Learning. Journal of Machine Learning Research, 12, 2411-2414.

(Tsoumakas, G., Katakis, I., 2007). Multi-Label Classification: An Overview. International Journal of Data Warehousing and Mining, 3(3), p.1-13, July-Sept 2007.

(WordNet, 2012). <http://wordnet.princeton.edu/>

(Younes, Z., Abdallah, F., Denoeux, T., Snoussi, H., 2011). A Dependent Multilabel Classification Method Derived from the k-Nearest Neighbor Rule. ; In Proceedings of EURASIP J. Adv. Sig. Proc. 2011.

(Zeni, N. Kiyavitskaya, N., Mich, L., Mylopoulos, J., Cordy, J.R., 2007). A Lightweight Approach to Semantic Annotation of Research Papers. ; In Proceedings of NLDB. 2007, 61-72.

(Zhang, M.L., Peña, J.M., Robles, V., 2009). Feature selection for multi-label naive Bayes classification. Information Sciences 179 (19), 3218-3229.

(Zhang, M.L., Zhou, Z.H., 2007). Ml-knn: A lazy learning approach to multi-label learning. Pattern Recognition, 40, 2038-2048.

List of Acronyms

ASCII	American Standard Character Information Interchange
L1 Tags	Level 1 of the Annotation Taxonomy
L2 Tags	Level 2 of the Annotation Taxonomy
MaLTAW	Machine Learning for Text Annotation Workbench
NP	Noun Phrase
PoS	Part of Speech
VP	Verb Phrase
XML	eXtensible Markup Language

REPORT DOCUMENTATION PAGE				<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY)		2. REPORT TYPE		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (include area code)