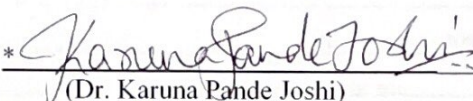




## APPROVAL SHEET

Title of Thesis: Multilingual Text Alignment For Cyber Security

Name of Candidate: Priyanka Ranade  
Master of Science, Information Systems 2019

Thesis and Abstract Approved: \*   
(Dr. Karuna Pande Joshi)  
(Assistant Professor)  
(Information Systems)

Date Approved: 04/11/19

NOTE: \*The Approval Sheet with the original signature must accompany the thesis or dissertation. No terminal punctuation is to be used.

## ABSTRACT

Title of dissertation: Multilingual Text Alignment  
For Cyber Security

Priyanka Ranade, Masters of Science, 2019

Dissertation directed by: Dr. Karuna Pande Joshi  
Department of Information Systems

Cybersecurity threats, exploits, and intelligence sources have evolved to be largely cross-regional over the course of time. Although the security community perpetually addresses this topic, its scope is continually stretching and introducing new areas of study. Particularly, an area of research that is relevant but heavily under-explored, is the use of multilingual open source intelligence in cyber operations. Open Source Intelligence (OSINT) in the form of text is scattered across major criminal networks, and is highly multilingual in nature. By aligning multilingual sources, the security community can tap into new pools of intelligence. Language alignment, can be achieved through the use of neural machine translation (NMT) systems. This thesis explores supervised and unsupervised methods in aligning multilingual open source intelligence sources without the use of third party engines. Although third party engines are growing stronger, they are unsuited for private security environments. First, sensitive intelligence is not a permitted input to third party engines due to privacy and confidentiality policies. In addition, third party engines produce generalized translations that tend to lack exclusive cyber

security terminology, which could be integral in attack discovery.

We address these issues and describe our system that enables threat intelligence understanding across unfamiliar languages. We create monolingual and multilingual word embeddings from open source intelligence data in two distinct languages, and derive a bilingual dictionary through both supervised and unsupervised methods. We then create a neural network based system that takes in cybersecurity data in a different language and outputs the respective English translation. We evaluate with traditional approaches, and through experimental applications.

# MULTILINGUAL TEXT ALIGNMENT FOR CYBER SECURITY

by

Priyanka Ranade

Dissertation submitted to the Faculty of the Graduate School of the  
University of Maryland, Baltimore County in partial fulfillment  
of the requirements for the degree of  
Masters of Science  
2019

Advisory Committee:

Dr. Karuna Pande Joshi, Chair/Advisor

Professor Anupam Joshi, Co-Advisor

Dr. Shimei Pan

Dr. Zhiyuan Chen

© Copyright by  
Priyanka Ranade  
2019



This is dedicated to my parents.

*“Listen to the intentions of God and do all things with love, and you will succeed.”*



## Acknowledgments

I would like to thank my advisor Dr. Karuna Pande Joshi for taking me under her wing and providing me the opportunity to further explore a concept that has always intrigued me. She has always made time to listen and provide me helpful advice and feedback, all with refreshing charisma that always inspired me. It has been an honor to have learned from her as a Masters student. My gratitude goes to my co-advisor Professor Anupam Joshi, who introduced me to the world of research while I was an undergraduate and cyber scholar at UMBC. His constant support, advice and encouragement has been a key factor in full circle of starting, advancing, and completing of this research. Collectively, I would like to thank my committee members Dr Shimei Pan and Dr. Zhiyuan Chen, who have both provided me their keen observations and feedback and have always made themselves available to me. As leaders in both AI and security, I am very fortunate to have utilized their advice to strengthen my work. Lastly, I would like to thank all of my friends in ACCL, KnACC, and Ebiquity Lab for the laughs, inspiration, and for letting me serve as a sponge to absorb knowledge from you. They taught me that being the least smartest person in a room, is the best place to be.

# Table of Contents

List of Tables	vi
List of Figures	vii
1 Introduction	1
1.1 The Semantics of Security . . . . .	1
1.2 Strengthening modern cyber defence systems . . . . .	2
1.3 Contribution . . . . .	3
1.3.1 Use Case . . . . .	4
1.4 Thesis Statement . . . . .	4
2 Overview and Related Work	6
2.1 Overview . . . . .	6
2.2 Background . . . . .	7
2.2.1 Social Media and Open Source Threat Intelligence Mining . .	7
2.2.2 Natural Language Processing . . . . .	7
2.2.3 Word Embedding Models . . . . .	8
2.2.4 Neural Machine Translation . . . . .	8
2.2.5 Multilingual Representation . . . . .	9
2.3 Related Work . . . . .	9
2.3.1 Text Analysis for domain specific tasks . . . . .	9
2.3.1.1 Vector Space Models . . . . .	9
2.3.2 Term Alignment in Vector Spaces . . . . .	10
2.3.3 Neural Machine Translation . . . . .	11
2.3.3.1 Cybersecurity understanding across multiple languages	11
2.3.4 AI Based Cyber Defense Systems . . . . .	12
3 System Design	13
3.1 Overview: Intelligence Translation Architecture . . . . .	13
3.2 Methodology . . . . .	13
3.3 Data Collection and Pre-Processing . . . . .	15
3.3.1 Collecting Open Source Intelligence data from Twitter . . . .	15
3.3.2 Pre-Processing . . . . .	16
3.4 Deriving Common Security Concepts . . . . .	18
3.5 Producing Monolingual word embeddings . . . . .	19
3.6 Bilingual Dictionary Creation . . . . .	21
3.7 Neural Machine Translation . . . . .	23
3.7.1 Architectural Details . . . . .	24
3.7.1.1 Encoder . . . . .	25
3.7.1.2 Decoder . . . . .	26
3.8 Implicit Learning with Crosslingual and Multilingual Embeddings . .	27

4	Evaluation	29
4.1	Word Embeddings . . . . .	29
4.2	Neural Machine Translation . . . . .	30
4.2.1	Cyber-Defense System Use . . . . .	34
5	Conclusion	37
	Bibliography	45

## List of Tables

3.1	LDA Concepts . . . . .	18
4.1	Evaluation Metrics . . . . .	32

## List of Figures

1.1	Multilingual Threat Intelligence Platform . . . . .	5
3.1	System Components . . . . .	14
3.2	Russian OSINT on Twitter . . . . .	16
3.3	Keywords . . . . .	17
3.4	Russian Cybersecurity Concepts for Topic Red . . . . .	19
3.5	Training Snapshot . . . . .	21
3.6	Synset . . . . .	22
3.7	Synset . . . . .	23
3.8	Intelligence Translation Network . . . . .	26
3.9	Training Snapshot . . . . .	27
4.1	Mean Average Precision . . . . .	30
4.2	Word Similarity . . . . .	31
4.3	Accuracy . . . . .	32
4.4	Loss . . . . .	33
4.5	Translation Samples . . . . .	34
4.6	Cyber Defense System Use . . . . .	35
4.7	RDF example. . . . .	36

# Chapter 1

## Introduction

### 1.1 The Semantics of Security

Information across political, cultural, and geographical boundaries is widely communicated over a global Internet. Today, we have a multilingual Internet where people converse in a variety of languages like English, Mandarin, Russian, Hindi, etc. [4]. Cyber threats in particular, originate from and are mitigated over a broad range of geographic regions. Although a significant amount cybersecurity web data is available, it is spread among major natural languages, decreasing interoperability between multilingual systems. This creates difficulty in employing strong cyber risk management across organizations worldwide. Specifically, amongst state actors or major criminal networks, it is likely that the threat information is in a language other than the language of the analyst.

Intelligence gathering spans an expansive geographic distribution. As a result, cybersecurity actors, both attackers and defenders, converse over *non-traditional sources* such as social media, blogs, dark web vulnerability markets, etc. in diverse languages. These non-traditional sources are becoming an important asset for threat intelligence mining [30] and many times are first to receive the latest intelligence about vulnerabilities, exploits, and threats [29]. The multilingual nature of these non-traditional sources is a potential hindrance for cyber-defense professionals, as

they might be limited by their knowledge of different languages. Despite this significant issue, the role of language in addressing cyber threats has been under explored. Multilingual understanding, adds to the many challenges security analysts continue to encounter. The security industry is heavily dependent upon the security analyst’s ability in using specialized experience to reason over the disparate pieces of intelligence data available on the web, in order to discover potential threats and attacks. A multilingual Internet needs a multilingual approach to cybersecurity.

## 1.2 Strengthening modern cyber defence systems

The abundance of cybersecurity web data has led to the use of AI/NLP based cyber-defense systems to help analysts extract relevant pieces of information that may constitute an attack. These systems need the ability to process multiple languages to keep up to date with the most current threat intelligence. While modern cyber defense systems have the ability to reason over disparate pieces of threat intelligence data on the web, we hope to create a defensive system that also understands various languages, by using the English language as a baseline. In our previous work, we developed *CyberTwitter* and *Cyber-All-Intel* [23, 24], systems that mine threat intelligence data from various sources, and automatically issues cybersecurity vulnerability alerts to users. This work extends these cyber-defense systems to a wider spectrum of potential threats, by mining threat intelligence data in a multitude of languages. These systems typically produce “cyber terminology representations” [23, 24] to categorize threat-related words, but only learn repre-

sentations for English. Consequently, if a certain threat is not gathered under a specific language, the system will not have a representation for it, even if it is a known threat in a different language. We use our multilingual threat intelligence system to align cyber terminology representations of different languages, expanding monitoring capabilities across the globe.

### 1.3 Contribution

In this thesis, we utilize word embeddings to align cybersecurity terminology in various languages. Primarily, we create a multilingual translation system that harnesses critical cybersecurity data derived from various natural languages to address the international nature of cyber attacks and assist in defensive cyber operations. Our system optimizes translations particularly for cybersecurity data. By also extending our system to use multilingual embeddings, we are able to transfer security knowledge from one language to another, without the overhead of producing bilingual dictionaries, as we did previously. Specifically, we investigate semantic representation of multiple languages with a corpus from Twitter, including threats and vulnerabilities in two languages, English and Russian. We build models to relate the vector space representations in the two languages to translate threat from Russian to English.



### 1.3.1 Use Case

## 1.4 Thesis Statement

“Structuring language alignment tasks on multilingual OSINT word embeddings, will allow us to create an in-domain neural machine translation model that yields more granular security term mappings in comparison to general translation engines.”

We will answer the following questions:

- Do cyber security terms differ enough across languages to provide dramatic impact to intelligence gathering efforts?
- Can we automatically align monolingual embeddings to discover relationships between multilingual cyber security words and reveal novel threat information in unfamiliar languages?

Our overall use case showed in Figure 1.1, utilizes embeddings created from Russian and English threat intelligence data. The embeddings help us understand security terms in Russian, by aligning semantically similar Russian cyber terms with their English counterparts. The system first begins by gathering relevant Russian threat intelligence data from sources such as Twitter. The data is then assimilated into a vector representation in order to bring semantically similar terms together [20]. The data is then fed into CyberTwitter, which converts the English representation of the Russian data into a machine understandable format defined using our UCO

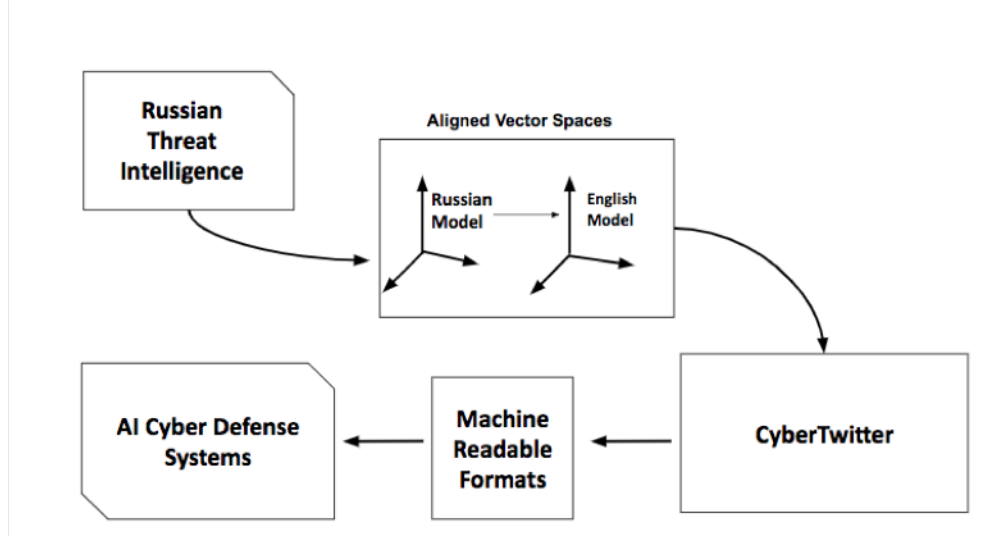


Figure 1.1: Multilingual Threat Intelligence Platform

Ontology [32] in OWL. This helps cyber-defense systems gain intelligence about threats mentioned in the Russian text. The acquired intelligence is then fed into an AI-based cyber defense system that generates conclusions from an accumulation of aggregated threat intelligence data.

An issue with directly converting threats in foreign languages to machine readable formats, is removing the security analyst from the threat inspection process. Providing analysts with raw translations help them reason over and expand upon a new landscape of threats and vulnerabilities. Our system aims to therefore, serve as an augmentation system that helps analysts divert full attention on their primary roles of analyzing and piecing in novel threat information.

## Chapter 2

### Overview and Related Work

#### 2.1 Overview

The internet hosts threat intelligence available in many languages, causing an overload of data to be processed. The ability to transfer knowledge of commonly understood languages to lesser known languages, certainly has the potential to grow information sharing capabilities. Refer to the pieces of foreign threat intelligence present below. Although translation systems are growing in popularity and have the potential to address this issue, many argue that even with grounded human truth they are not as reliable as human experts analysts. As expected, there seems to be a low degree of human translators to handle the superfluous amount of potentially useful multilingual cyber security data. To combat this issue, we first create a domain-specific translation model that takes the help of Russian cyber security analysts to train Russian text to output its English counterparts. We compared the accuracy of our model against third party translation systems, which are state of the art but not preferred in sensitive security environments due to their scalability issues as well as, privacy and confidentiality limitations. Our comparison was mainly used to determine if we can output more cyber security specific translation. Although we got seemingly good accuracy and usability, our next step was to make our system could more useful with the creation of crosslingual embeddings. These crosslingual

embeddings can also be used as input to other tasks such as question and answering systems, and even the production of domain-specific multilingual embeddings.

## 2.2 Background

The methodology and architecture details are described in Chapter 3. Before we discuss the details of system, we will briefly introduce the key concepts used in this thesis.

### 2.2.1 Social Media and Open Source Threat Intelligence Mining

The Intelligence Community heavily relies on openly available information on the internet to identify potentially malicious events. This information is commonly known as Open Source Intelligence, and is gathered from sources such as online newspapers, blogs social-networking sites, etc. Due to its breadth and easy access, OSINT has emerged as an important aspect of the threat inspection process security analysts perform. Social media in particular, delivers human sentiment and opinion, trend analysis, and real time worldly event alerts, which together have proven to reveal and provide threat indicators to the security community.

### 2.2.2 Natural Language Processing

Natural Language Processing (NLP) is a field within computer science, that enables computers to understand human language. NLP includes syntactic tasks such as part of speech tagging [20, 19] and semantic tasks such as machine translation [22].

A standardized method to capture semantic properties of words is the production of "Word Embeddings", described in the next section.

### 2.2.3 Word Embedding Models

Word embeddings are popular NLP architectures used for a variety of tasks from recommendation systems to neural machine translation systems. Word embeddings are neural networks, that represent words as real numbers in a continuous vector space. Word embeddings can be produced through various algorithms like CBOW[], Skip-Gram, and Latent Semantic Analysis [LSA]. In this work, we utilize the CBOW algorithm to produce monolingual embeddings, vectors compromised of one natural language and cross-lingual embeddings, or a representation of two natural languages in one vector space.

### 2.2.4 Neural Machine Translation

Word embeddings are principle architectural components in Neural Machine Translation (NMT) tasks. Generally, semi-supervised NMT models have used a Sequence to Sequence architecture, in which a decoder learns a generalized semantic encoding of a source language, and outputs a target translation, moving away from phrase-phrase translation. These models make use of a bilingual dictionary of source to target language. More recent research has moved towards unsupervised NMT through the use of automatic dictionary generation with multilingual word embeddings.

### 2.2.5 Multilingual Representation

The rise of linked data has stimulated semantic information publication on the web by enabling vast opportunities to connect datasets through machine-readable formats. Moving towards a Semantic Web also presents the ability to publish knowledge in various natural languages, like Russian and Chinese. Currently, there are over 100 languages represented on the internet. W3Techs estimated English and Russian ranking as the top two of most represented languages, after surveying the top 10 million websites on the web in 2017.

## 2.3 Related Work

In this section, we present related work on the vector space model uses, neural machine translation, AI-based cybersecurity systems, and cybersecurity understanding across different languages.

### 2.3.1 Text Analysis for domain specific tasks

Text analytics has been utilized in areas such as information retrieval [16], machine translation [13], and topic detection [18]. These areas are especially useful for domain specific tasks such as cybersecurity.

#### 2.3.1.1 Vector Space Models

Vector Space Models, or word embeddings, have been used in Natural Language Processing. Words are embedded in a continuous vector space such that,

words that appear in the same contexts are semantically related. One method that generates embeddings based on word co-occurrence is word2vec [20, 19]. Mikolov et al. [19], showed that proportional analogies can be solved by finding the vector closest to the hypothetical vector. Embeddings have also been utilized in other areas such as word sense disambiguation [3], semantic search [31], and discovering inter-linguistic relations in machine translation studies [22].

Wordnet [21] is a human curated lexical database that groups together synonyms in the English language. Many other versions of Wordnet have been produced, such as ArabicWordNet and ChineseWordnet [27]. These lexical databases are often times used to aid lexical and term alignment.

### 2.3.2 Term Alignment in Vector Spaces

Analogical Relationships are often times utilized to aid term alignment in vector spaces. Term alignment is known as statistically finding correspondences between words in different groups [5]. Plas et al. [17] utilize automatic word alignment to find translations from Dutch to one or more target languages. Similarly, Brown et al. [28] aligned sentences with their translations in two parallel corpora, consisting of French and English. Yang et al. [25] show how the pattern of the context from word embeddings help to align similar word pairs in other languages. Piantra et al. [9] created MultiWordnet, an aligned multilingual database curated to produce an Italian Wordnet, by aligning synonyms in Italian to EuroWordNet. Niemann et al. [8] aligned WordNet synonym sets and Wikipedia articles to group

article topics based on synonyms.

### 2.3.3 Neural Machine Translation

Word embeddings have aided in a diversity of machine translation tasks. Neural machine translation typically operates through the encoder-decoder-attention architecture [7]. More recently, bilingual word distributions have been trained using unsupervised methods such as Latent Dirichlet allocation (LDA) and Latent semantic analysis (LSA) to aid machine neural translation [14]. Lample et al. [12] trained word embeddings from monolingual data and utilized external and internal vectors as input for the network utilized to train unfamiliar instances of words. In terms of semantic translation tasks, Hill et al. [10] show that translation-based embeddings work better in applications that require concepts organized according to similarity

#### 2.3.3.1 Cybersecurity understanding across multiple languages

Cybersecurity terminology definitions differ across cultures and languages. The Department of Homeland Security started developing multilingual resources, to help link cybersecurity understanding across international governments [6]. Klavens et al. [15] outlines the importance of linguistics in the domain of security and claims language analysis propels understanding of communication between cyber-crime activist groups, filtering relevant data collection, and understanding the intention behind the words.



### 2.3.4 AI Based Cyber Defense Systems

The use of social media in threat intelligence mining, provides a new interface between the public and the Intelligence Community. Twitter data in particular, is seen as a reliable OISNT resource due to its real time nature during high impact events, such as terrorist attacks [1]. Mittal et al. [23] developed CyberTwitter, a threat intelligence framework that utilizes twitter data to automatically issue security vulnerability alerts to users. Similarly, the Cyber-All-Intel system collects OISNT data, stores it in a cybersecurity corpus, and utilizes word vectors for cybersecurity term similarity searches [24].

## Chapter 3

### System Design

#### 3.1 Overview: Intelligence Translation Architecture

In this section, we describe our data collection methods, vector space generation, alignment techniques and neural machine translation framework. We first create a multilingual cybersecurity corpus that contains tweets about threats and vulnerabilities in various languages. In this paper, to create a proof of concept, we focus on English and Russian. We investigate semantic alignment of both languages through implicit and explicit learning. Using the collected corpus, we then produce English and Russian vector embeddings. Once we create the embeddings, we align both vector spaces utilizing an alignment database. Once the spaces are aligned, we are explicitly translate of Russian threat intelligence to English. We then utilize a linear transformer to represent both languages in a single vector space as an experimental use case.

#### 3.2 Methodology

This work’s primary goal is to configure different methods in aligning and comparing cyber security terminology and understanding across any language. Figure 3.1 below shows the major components of this thesis work. Through both

semi-supervised and unsupervised learning, as well as multiple use cases to test the system below, we conclude that in-domain modeling yields more granular security term mappings in comparison to general translation engines. The details of each component in Figure 3.1 are explained throughout the rest of this chapter. We first create a multilingual cyber security corpus that contains tweets about threats and vulnerabilities in various languages. In this paper, to create a proof of concept, we focus on English and Russian. We then create a topic model to derive cybersecurity concepts in both the Russian and English languages. These topics will later aid us in creating the alignment database discussed in Section 3.6. We then use the corpus to create monolingual English and Russian vector space embeddings. Once we create the embeddings, we align both vector spaces utilizing an alignment database. Once the spaces are aligned, we are able to undertake semantic translation of Russian cyber threats and vulnerabilities to English.

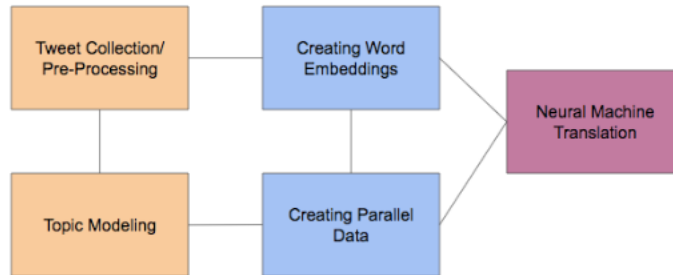


Figure 3.1: System Components

### 3.3 Data Collection and Pre-Processing

In this section, we describe our data collection methods, and crosslingual OSINT corpus. We start by describing our data Collection methodology 3.3. We then explain methods in discovering cyber security concepts in each language 3.4. Next, we create English and Russian word embeddings from the tweets 3.5. After, we create a bilingual dictionary 3.6 later used in our neural machine translation system 3.7. Finally, we use a linear transformer to align both vector spaces into one embedding using Facebook’s FastText algorithm 3.8.

#### 3.3.1 Collecting Open Source Intelligence data from Twitter

We collect data through the Twitter streaming API. We explore open source intelligence data available through twitter due to its real time nature, as well as its tendency to compose information from various other sources onto one platform. Through twitter, we were able to naturally interface with security bloggers, security organizations, product companies, as well as everyday users who discover vulnerabilities. An example of open source intelligence available on twitter is shown in Figure 3.2

We collect tweets upon three major categories. The first two categories compromise keywords significant to each language. These keywords were suggested by multilingual cyber security domain experts [26] and various security analysts. Collecting data using the keywords shown in Figure 3.3 gives us a direct interface to Russian cyber colloquialisms. For example, the tweet depicted in Figure 3.2, reveals

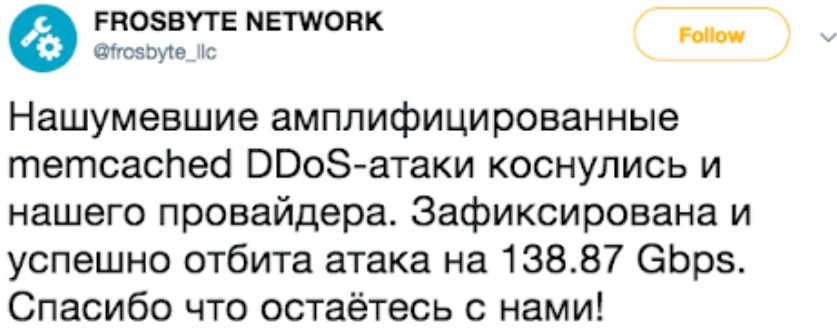


Figure 3.2: Russian OSINT on Twitter

a regional-specific DDoS attack, to threat analysts outside of Ukraine. The third category, included the same words back-translated to the corresponding language. The back translation helped us conclude that while some security terms retain the same meaning, others are largely different. This is more clearly explained in Section 3.7.1. In this effort, we hoped to find the extent to which security terms differed across the two languages. We use the Twitter API language capabilities to detect tweet language through a flag (en=English, ru=Russian). Setting this flag provides us the ability to collect data in both languages.

The data is stored and separated by language in MongoDB. MongoDB is a NOSQL database that later helped us in storing and creating, our bilingual dictionary described in Section 3.6.

### 3.3.2 Pre-Processing

After creating the twitter corpus, we pre-process the tweets by language. Twitter data is highly unstructured, containing special characters, emoticons, as well as

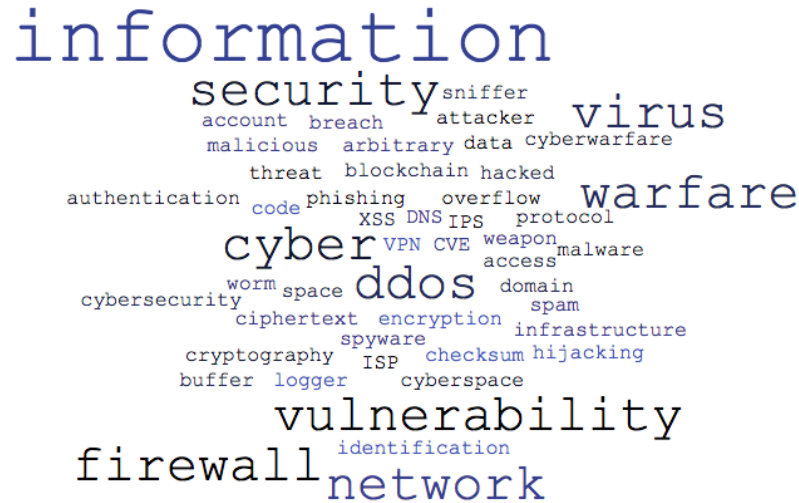


Figure 3.3: Keywords

slang terminology. The pre-processing step was especially crucial for this thesis in order to retain the most significant information amongst the noise. For both languages, we first tokenize the words by space. We then, remove all stop words and stem the words to their root forms. We later lemmatize the words in order to. Lastly, we remove all special characters except hashtags due to their heavy importance in a tweet. The Russian and English tweets use their own tokenizers, stemmers and lemmatizers specific to the language. We utilize the plunkt library for the Russian corpus. Hashtags are powerful mechanisms in tweets, in that they have the power to summarize a tweet with a few major key words. These hashtags were important for us in our initial analysis. We were able to compare one to one major words across two languages, before even creating embeddings to analyze the sentences. For our translation efforts in Section 3.7.1 we take the hashtags out for modeling purposes. After pre-processing our corpus size reduced from 2G of data, to 424,928KB of data.

Concept	Definition	Example
Red	Systematic Attack Verbs	spoof,thwart,exploit
Blue	Preventative Terms	protect, warning, report
Technical	Explicit Attack Details	rootkit, dns, xs
Political	Regional Conflict	russia,strategy,war

Table 3.1: LDA Concepts

### 3.4 Deriving Common Security Concepts

In this section we describe our topic model for deriving and analyzing concepts for both the Russian and English tweets. We apply Latent Dirichlet Allocation separately for the Russian and English tweets, and later compare the results. Latent Dirichlet Analysis is a probabilistic topic model used to derive classes of a document based on. This model assumes a Dirichlet Prior over the topics. For the Russian tweets, we derived 135 concepts overall, and for the English tweets, we derived 72 concepts. Through the help of a Russian speaker, we sifted through the concepts to manually extract first, security related classes, and secondly similarities in classes between both languages. Examples of similar security concepts we derived are displayed in Table 3.4 below. We found the concepts derived related directly to both offensive and defensive sides of the security community. The Red concept classified attack vector terms, while the Blue concept, classified defense terminology. Figure 3.4 represents a snapshot of Red security topics related to IOT devices.

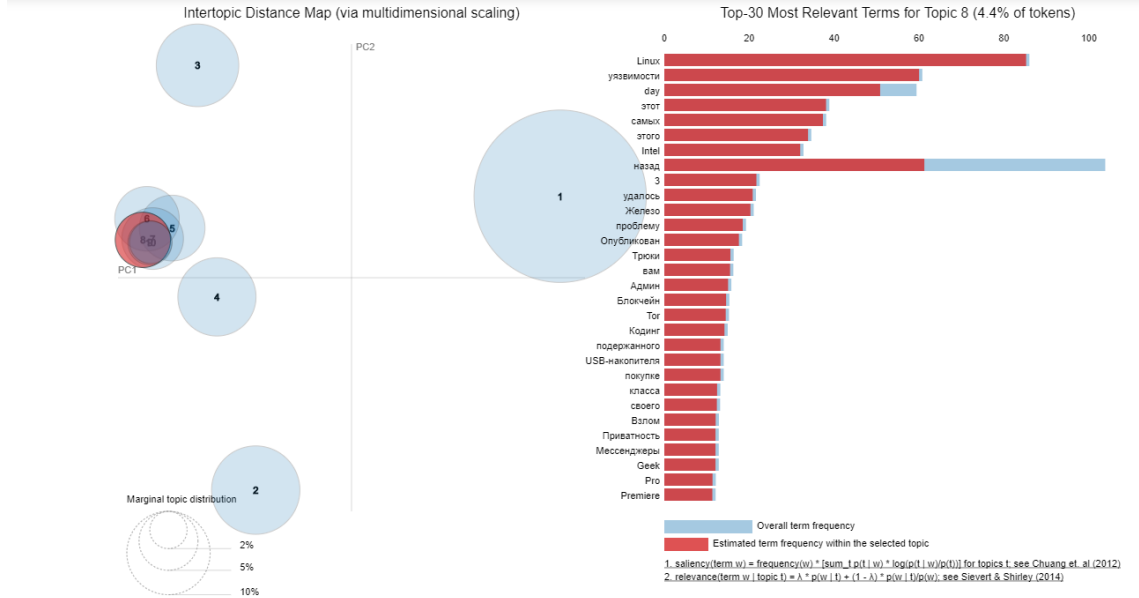


Figure 3.4: Russian Cybersecurity Concepts for Topic Red

### 3.5 Producing Monolingual word embeddings

In order to learn mappings between security terms inside of one language, we develop separate vector space models for the English tweets and the Russian tweets. We are able to analyze semantic similarity of words in the same language space by creating monolingual embeddings. Word embeddings composed of one language are called monolingual embeddings, while embeddings composed of two are crosslingual embeddings. More information on the unsupervised crosslingual embeddings we developed with Facebook FastText can be found in Section 3.8. We generate English and Russian word embeddings with Word2Vec. Word2Vec is a generalized package of word vectors that can be used in multiple applications. In our application, we train on a corpus of Russian and English tweets to find a generalized representation of the cyber security related words.



Word2Vec comes with two different models, the Skip Gram and the Continuous Bag of Words (CBOW) model. We use CBOW to represent our collection of tweets as a vector. Words in the embedding space are semantically similar if grouped together around the same neighborhood. For example, in our English model, words like malware,iot, network,ddos,community,etc. will be clustered together.

We import word2vec through the Gensim python library. The word embeddings for each language are multidimensional and have 300 dimensions each. The more dimensions a word embedding has, the more generalized the model is, and hence the more accurate the model is. We chose 300 dimensions, as it was the largest number of dimensions we could implement, without making the system too computationally expensive. The minimum word count for the model is 2. This means, the words with an occurrence less than 2 will be ignored. We chose a lower word count, seeing that more rare threat words will only appear once or twice, but still need to be retained in the vocabulary. We initialized two workers running parallel in one CPU core. Our context window, is 10, meaning the model blocks 10 words to process at a time. We set the down sampling size to 1e-3, to reduce frequent word appearance while training. Past research has shown that a strong range is 0 - 1e-5. Lastly, we set 1 seed to serve as a deterministic random number generator in order to pick which parts of the corpus to produce into vectors. The embeddings train for 35 epochs.

Figure 3.5, depicts a 20th iteration training snapshot, of Russian words that start appearing near “DDoS”, a type of cyber security attack. These monolingual embeddings are later used in the neural machine translation system we develop,



Figure 3.5: Training Snapshot

described in Section 3.7.1.

### 3.6 Bilingual Dictionary Creation

In this section we describe the methods used to create the Bilingual English to Russian Dictionary, as well as its two purposes in this research. Its first purpose, is to serve as an evaluation task for the word embeddings produced in Section 3.5. The evaluation process is described in Section 4.1. The second purpose, is to serve as parallel data that is used in neural machine translation training described in Section 3.7.1.1 and 3.7.1.2.

In order to create relationships between English and Russian cybersecurity

words, we created a dataset to align the English and Russian vector embeddings. An alignment in our system means creating true positive mappings of Russian cybersecurity terms to their English counterparts. We derived cybersecurity synsets for the Russian and English vocabulary embeddings, created in Section 3.6. These cybersecurity synsets include contextually similar words to each vocabulary word in the Russian and English vector spaces. We emphasize that, when we say contextually similar words, we bring together cybersecurity terms in the same word sense. The lexical database Wordnet [22], groups similar words into sets of synonyms called synsets. WordNet does not support the Russian language. We found a similar lexical database called Wiki-Ru-Wordnet, specifically for the Russian language. We utilize the English synsets provided by WordNet, and the Russian synsets provided by Russnet to create our cybersecurity synsets. An example of a cyber synset we derived is shown in Figure 3.6.

**Set\_intrusion** ['invasion.v.01', 'disruption.v.02', 'infringement .v.03']  
**Set\_интрузия** ['внедрение.v.01', 'приглашения.v.02', 'вторжение.v.03']

Figure 3.6: Synset

The process for this creating the aligned database is shown in Algorithm 1. We converted the Russian and English vocabularies of each synset into a list comprehension and derived synets for each member of the list, and stored it in a dictionary. Each vocabulary word serves as a key that maps to many values. In this case, as each synset is derived for a vocabulary term, the synset is appended to a key value.

We tasked two native Russian speakers, who served as annotators, to manually

---

**Algorithm 1** Synset Dictionary

---

```
REQUIRE Wordnet as wn
L ← VocabularyList
  for all v ∈ L
D ← Dictionary
  for all vocab-synset pairs (v,s) ∈ D
for v in L:
  D = { }
cyber_sets = wn.synsets( word)
for cyber_set in cybersets:
  D["v"].append(cyber_set)
```

Figure 3.7: Synset

verify the quality of cyber security synsets produced. We use the Cohens Kappa to compute the inter-annotator agreement, and keep only those cyber security synsets that scored higher than 0.66.

The annotators confirmed that the synsets in Wordnet, and the synsets in Russnet, were not only similar on a translation level, but also semantically similar in a cultural context.

### 3.7 Neural Machine Translation

In this section, we describe our intelligence translation framework that takes as input a Russian tweet and outputs its respective English translation. This model is an example of an in-domain architecture, which translates data specifically for cyber security. We use the cyber security embeddings developed in Section 3.5 as well as the bilingual dictionary containing cyber security synsets, as part of the

framework.

Our intelligence translation architecture is shown in Figure 5. We implement a standard a encoder-decoder network, which is a dual Recurrent Neural Network (RNN). The encoder serves as an input RNN and the decoder, serves as the output. The encoder-decoder architecture projects the input Russian word to be translated into the English embedding space, by returning words with a representation closest in the English vocabulary.

The encoder-decoder network is implemented using a sequence-to-sequence architecture [8], [32]. The encoder decoder network is able to process past and future words in a sequence, and is also able to map an input sequence to an output sequence of a different length. This is important to note because the translation will almost always contain a different number of words than the input. If we used a a typical RNN rather than a bidirectional RNN, our model would return one hidden state per input, and get one translation per input, and the output length will be the same as the input length at each point in time. The details of each part of the network are described below.

### 3.7.1 Architectural Details

In this thesis, we use the Keras seq2seq, an implementation of the encoder decoder network. Our architecture is displayed in Figure 3.8. We start by taking raw input in the form of a Russian tweet. We then create a compressed vectorized representation of the tweet through the first encoder RNN. The decoder RNN pro-

duces a translation from the compressed vector representation of the input. The encoder and decoder utilize a Long Short Term Memory (LSTM) cell [8]. More information on the encoder and decoder states is explained in Section 3.7.1.1 and 3.7.1.2 respectively. The model uses pretrained word embeddings produced in Section 3.5. Due to the high accuracy of the embeddings we trained previously, we reuse them, to reduce training time. More information on accuracy can be found coming in Section 4.2. We initialize Russian embeddings as the input in the encoder state, and the English embeddings as the output of the decoder state. We utilize the cybersecurity synsets, from Section 3.6 as well as the very popular Tatoeba aligned en-ru sentence data as our parallel data. We train on 8000 samples and validate on 2000 samples. Our hyperparameters were set as, batch size = 64, epochs = 100, latent dimensionality = 256, sample number = 10,000. In the hidden layer, we have one dense layer, with a softmax activation function, which allows the model to learn a mapping from the Russian vector representation, to the English vector representation. The encoder, takes in Russian words and maps them to their respective vector representations. The decoder then, creates a translation of the input word and generates its predicted aligned semantic English embedding.

### 3.7.1.1 Encoder

The encoder operates like a classic RNN described in Section 3.7. We first pass in the raw input, and the model returns a series of states (h and c). The encoder does not produce any predictions, and functions to only train the model.

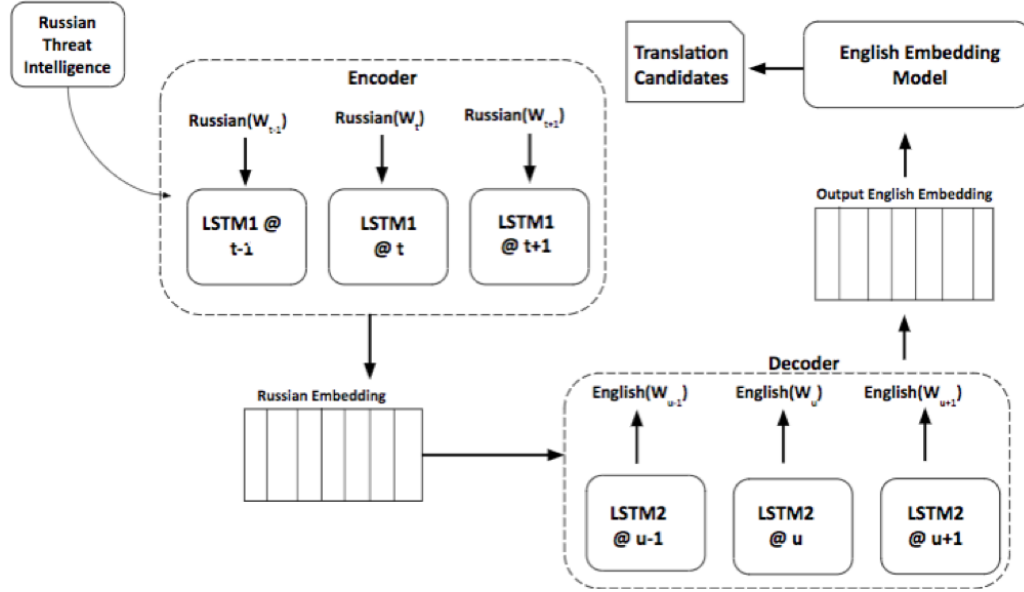


Figure 3.8: Intelligence Translation Network

The LSTM retains the last state of the sequence  $(h, c)$ . The encoder gives us a function  $h(t)$  which is considered the thought vector. The thought vector is the compressed vectorized representation of the input sequence and is passed to the Decoder state.

### 3.7.1.2 Decoder

The decoder utilizes another LSTM cell. The thought vector from the encoder is passed into the decoder, making them the same unit size. We then pass in the first token of the sequence along with a start of sentence token that pads the input.

(insert formula)

Using the previous state  $(h_1)$  and the first token  $(x)$ , the model can predict  $y_1$ . We then take the argmax of probabilistic  $y_1$ , in order to return the most likely

word in the target language. A training snapshot of the model is shown in Figure 3.9 below.

```
Epoch 95/100
8000/8000 [=====] - 129s 16ms/step - loss: 0.1137 - acc: 0.9554 - val_loss: 2.5675 - val_acc: 0.7252
Epoch 96/100
8000/8000 [=====] - 128s 16ms/step - loss: 0.1128 - acc: 0.9558 - val_loss: 2.5744 - val_acc: 0.7241
Epoch 97/100
8000/8000 [=====] - 128s 16ms/step - loss: 0.1120 - acc: 0.9559 - val_loss: 2.5986 - val_acc: 0.7246
Epoch 98/100
8000/8000 [=====] - 128s 16ms/step - loss: 0.1123 - acc: 0.9554 - val_loss: 2.5786 - val_acc: 0.7252
Epoch 99/100
8000/8000 [=====] - 128s 16ms/step - loss: 0.1114 - acc: 0.9553 - val_loss: 2.5961 - val_acc: 0.7233
Epoch 100/100
8000/8000 [=====] - 128s 16ms/step - loss: 0.1113 - acc: 0.9551 - val_loss: 2.5910 - val_acc: 0.7249
```

Figure 3.9: Training Snapshot

### 3.8 Implicit Learning with Crosslingual and Multilingual Embeddings

In previous sections, we describe semi-supervised methods for aligning the vocabularies of the Russian and English vector spaces. In this section, we describe the unsupervised approach we used to align both embeddings in one space. We use Facebook’s open source MUSE library to perform the alignment. MUSE provides a linear transformer that learns a crosslingual mapping using adversarial training and iterative Procrustes refinement. Procrustes refinement is used to map two configurations that have different dimensions. Procrustes analysis achieves this by using one configuration as a mapping and fitting the second one to it through some kind of transformation. From our results, we found 78 percent correspondence of the



analogy assessments with the unsupervised mappings of Facebook’s algorithm. We calculated this by taking the vocabulary of the English language, and finding the most similar words in both Russian and English, for each vocabulary word. We employed the test mentioned in Section 4.1 for the Russian and English words. Due to the high correspondence, we were able to show that this project is feasible in the future, without heavy resources such as the manual effort that we required for the neural machine translation system. We We utilize this algorithm as a major baseline in assessing the embedding alignments we made from a systematic view.

## Chapter 4

### Evaluation

#### 4.1 Word Embeddings

We utilize an analogy assessment to evaluate the English and Russian monolingual and multilingual embeddings. Analogy assessments are state of the art evaluation subtasks used to test word similarity of target words, to its neighbors. We first use word2vec’s analogy task which utilizes the cosine similarity to assess semantic similarity. In order to assess the relationship one word to another, the task takes the vector for word (a) and negates the vector of word (b) and finally adds the vector of word (c). The output of this would be the most similar, or analogous word. An example of a similarity task done for the word "iot" for the Russian monolingual space is shown in Figure 4.2.

We also employ our own assessments to better evaluate our models for domain specific cyber security tasks. We employ Mean Average Precision to assess the presence of indomain cybersecurity words and their synonyms in our embedding vocabularies. To do this, we map the the most similar words of each vocabulary list as shown above, to the synonym set database described in Section 3.6. By storing the analogies of the words in MongoDB, we were able to query value pairs for each of the vocabulary words, and compare them with regular expressions to the most similar words derived from word2vec most similar function.

English	47%
Russian	44%

Figure 4.1: Mean Average Precision

## 4.2 Neural Machine Translation

In this section, we describe our experimental setup and evaluate our intelligence translation system.

We first evaluate our encoder-decoder architecture through an accuracy metric and a BLEU (Bilingual Evaluation Understudy) score (see Table 1). The accuracy metric computes the percentage of times that predictions match labels. BLEU scores, are standard metrics for evaluating a generated translation to a reference word [11]. We use NLTK’s `sentencebleu()` function and provide a list of reference and candidate sentences, given to us by our annotators. The score is generated by counting the matches of n-grams in the reference sentence to n-grams in the candidate sentence. An accuracy above “60%’ and a BLEU Score between “15 and 36” is considered robust [11]. Accuracy, Validation and Loss metrics over each epoch are shown in Figure 4.3 and 4.4 respectively.

We measure the precision of our translations by checking a randomly generated sample of the output against Google Translate<sup>1</sup>. We proved that our system produces more effective translations for the security domain. We extracted 1000

---

<sup>1</sup><https://translate.google.com>

```

russian2vec.most_similar("iot")

c:\users\ebiquity\appdata\local\programs
deprecated `most_similar` (Method will b
    """Entry point for launching an IPytho

[('осуществить', 0.9997989535331726),
 ('Украины', 0.9997894763946533),
 ('которого', 0.9997826218605042),
 ('мессенджера', 0.9997802972793579),
 ('многих', 0.9997791051864624),
 ('является', 0.9997783899307251),
 ('Signal', 0.9997783899307251),
 ('специалистами', 0.9997783303260803),
 ('второй', 0.999777615070343),
 ('версию', 0.9997747540473938)]

```

Figure 4.2: Word Similarity

randomly selected tweet translations and compared the output against the Google Translate API. We check our translations against the ones provided by Google Translate, both syntactically and semantically. On evaluating 1000 random samples, there was a 64.3% syntactic correlation (BLEU-2) between the two systems, showing that our system is comparable to a state of the art architecture, therefore showing reliability in translations. We further evaluated the 357 samples that were not syntactically equivalent and tasked two security analysts to manually evaluate the semantic meanings of the translation outputs. We found that of the 357 outputs, 349 were semantically similar, but not syntactically similar to the Google translation, showing

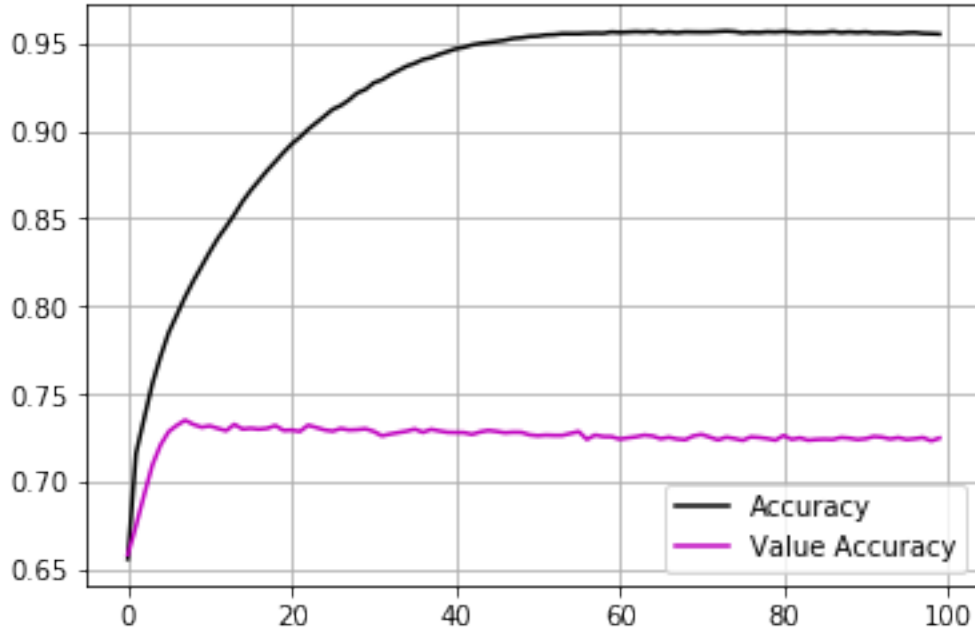


Figure 4.3: Accuracy

Measure	Value
Accuracy	95.22%
BLEU Score	.284

Table 4.1: Evaluation Metrics

97% semantic relevance. We define semantically similar as translations that do not meet level BLEU-2 , but generate the same underlying meaning. The annotators concluded our translations are preferable through a security perspective, in that they proliferate terms unique to the security industry. The commercial translation services are generalized while our system is domain specific. These security specific translations can be attributed to the architecture of our model, that utilizes a specialized aligned database made with relevant cyber security mappings. Examples of

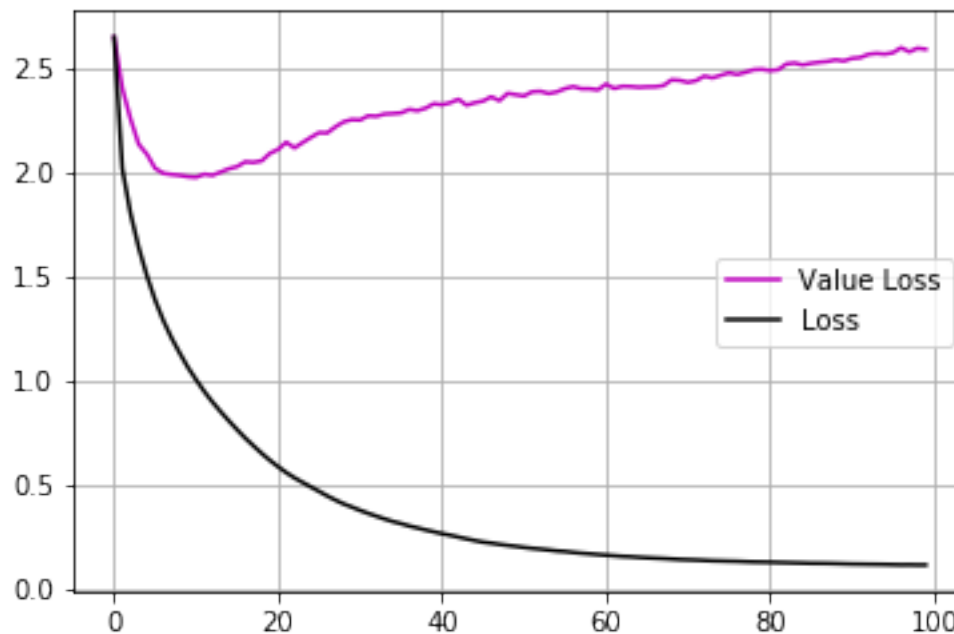


Figure 4.4: Loss

unequal but semantically similar translations in our system and Google Translate are listed in Figure 4.5. In example 1, malware registers more with a security analyst than malicious programs. In example 2, the Google Translate system translated the relevant Russian text as *spylair*, while our system gives the correct translation as *spyware*. These are clear instances in which our translation will provide more relevant and direct intelligence for a security professional. Another benefit that our system provides is that it can run independently in secluded operational settings. A security analyst may not be able to input their sensitive data into third party platforms due to privacy, security, and confidentiality policies.

Original Russian Tweet	Intelligence Translation System	Google Translate
Вредоносные программы установлены на устройствах китайских производителей	Malware installed on devices of Chinese manufacturers	Malicious programs are installed on devices of Chinese manufacturers
Разработчики убирают шпионское приложение из-за протестов игроков	Developers clean spyware application because of player protests	Developers clean spyware due to protests players
Positive Technologies: хакнуть процессоры Intel можно через USB порт и отладочный интерфейс	Positive Technologies: Hack Intel processors with a USB port and a debug interface	Positive Technologies: Intel processors can be hacked via a USB port and a debugging interface
При открытии сайта Минэнерго высвечивается только красная страница, на которой написано что сайт зашифрован	Opening of the Ministry of Defense page, displays encrypted red page.	When the website of the Ministry of Energy is opened, only the red page is displayed, on which it is written that the site is encrypted

Figure 4.5: Translation Samples

#### 4.2.1 Cyber-Defense System Use

Web based unstructured, textual sources such as Twitter, Reddit, blogs, dark web forums, etc. provide a rich multilingual source of information about cyber threats and attacks. In addition to providing details of existing attacks, such sources (especially the dark web) can serve as advance indicators of attacks in terms of discussions around newly discovered vulnerabilities. This information is available in textual sources traditionally associated with Open Sources Intelligence (OSINT), as well as in data that is present in hidden sources like dark web vulnerability markets.

The intelligence translation system that we discuss in Section 3.7.1 will help us automate this process by taking data from a variety of multilingual sources, extracting, representing and integrating the knowledge present in it as embeddings and knowledge graphs, and then use the resulting artificial intelligence systems to provide actionable insights to SoC professionals. Figure 4.6 showcases our pipeline,

which takes in Russian threat intelligence and stores it in as a VKG structure [25].

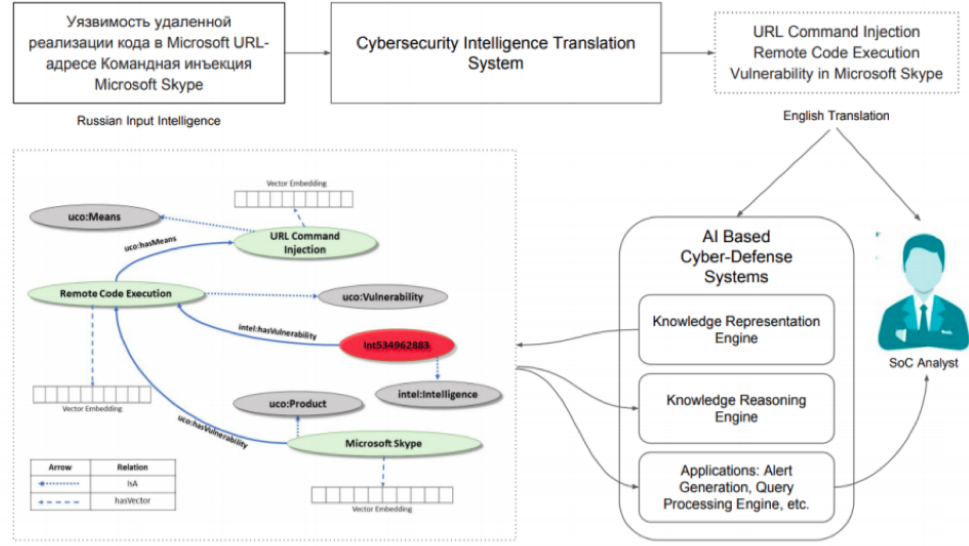


Figure 4.6: Cyber Defense System Use

Two such systems that we have developed in the past are CyberTwitter [24] and Cyber-All-Intel [25]. The systems store threat intelligence in a knowledge representation that can be used by AI based cyber-defense systems (See Figure 7). Such systems generally have a knowledge representation engine, a reasoning engine, and few applications like an alert generation system, recommender system, query processing system, etc.

The knowledge representation system, converts input threat intelligence (usually in a textual format) into a machine readable format. In our system we represent it in RDF2 ,with cybersecurity domain knowledge provided by the Unified Cybersecurity Ontology (UCO) [34]. The intelligence ontology [24] provides information about the intelligence domain. We also include specific conceptual embeddings for security concepts in our threat representation format [25]. The knowledge reasoning



part of the system provides domain specific reasoning capability generally encoded as logical rules by a domain expert. The applications and the reasoning engine generally use the machine readable representation to provide specific functionality. Figure 7 also provides the graph structure for the translated English intelligence: URL Command Injection Remote Code Execution Vulnerability in Microsoft Skype. Figure 4.7 provides the RDF representation for the same intelligence.

```

@prefix uco: <http://accl.umbc.edu/ns/ontology/uco#> .
@prefix intel: <http://accl.umbc.edu/ns/ontology/intelligence#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dbp: <http://dbpedia.org/resource#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

<Int534962883> a intel:Intelligence ;
intel:hasVulnerability <remote_code_execution> ;

<command_injection> a uco:Means .

<Microsoft_Skype> a uco:Product ;
uco:hasVulnerability <remote_code_execution> ;
owl:sameAs dbp:Skype .

<remote_code_execution> a uco:Vulnerability ;
uco:affectsProduct <Microsoft_Skype> ;
uco:hasMeans <command_injection> ;
owl:sameAs dbp:remote_code_execution .

```

Figure 4.7: RDF for textual input “URL Command Injection Remote Code Execution Vulnerability in Microsoft Skype”. Also, *owl:sameAs* property has been used to augment the data using an external source ‘DBpedia’ [2].

## Chapter 5

### Conclusion

In this paper, we described the design, implementation, and evaluation of a multilingual threat intelligence alignment system. We first collect Open Source Intelligence data from twitter in two natural languages: Russian and English. We then, derive common cybersecurity concepts in each language with LDA. After, we create a domain specific cybersecurity term alignment dictionary that serves as training and test data for the in domain neural machine translation system we develop, as well as an evaluation task for the cybersecurity embeddings we produce. The system uses Russian and English word embeddings created from cybersecurity data, an aligned cyber term database, and a LSTM based neural machine translation architecture, to translate cybersecurity text from Russian to English. With the help of Russian speaking cyber analysts, we created an alignment database by generating synonyms for the Russian and English corpus vocabularies, along with their respective translated Russian and English words. We utilize this database in neural machine translation, where we use an encoder-decoder architecture to map unfamiliar Russian cyber inputs to their English counterparts. We show that our model not only has high syntactic correlation to third party translation systems, but also registers prevalent cybersecurity terms in translation better than third party engines. We extend third party translation systems by creating a domain specific

model that can provide more pertinent intelligence for an analyst. Our system can be utilized in private operational settings that do not permit the use of third party applications when dealing with sensitive intelligence data. We also align both embeddings unsupervised through Facebook’s linear transformer.

A weakness of our system, is the requirement of a cybersecurity rich alignment to train the model. Although we derived a Russian and English cybersecurity synonym sets in this proof of concept, it is an expensive task that will take dispersed effort across the linguistic and security communities, to derive across many other languages.

In order to create more mappings for cyber terms across other languages like, Mandarin, Cantonese, Portuguese, Arabic, Hindi, etc. future research can include automatic creation of multilingual cyber alignment databases. We can also consider transferring knowledge from languages with an abundance of intelligence to other unknown languages with no or few alignments through multilingual embeddings. We expect that aligned cyber embeddings across many languages can promote international incident response collaboration. In addition, a highly unexplored area is analyzing leetspeak in darkweb networks. Leetspeak is widely known as code language that criminals use in order to openly communicate systematic attack details, without becoming detected or understood by cyber defense systems. Another future application could be applying methods to translate and represent leetspeak, against natural languages known to us.

## Appendix

### Code for this Thesis

```
#Referenced the gensim word2vec documentation and the following repos for guidance:
#https://github.com/l1Source11/

from __future__ import absolute_import, division, print_function

import codecs

import glob

import logging

import multiprocessing

import os

import pprint

import re

import nltk

import gensim.models.word2vec as w2v

import sklearn.manifold

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import seaborn as sns

get_ipython().run_line_magic('pylab', 'inline')

logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)

nltk.download("punkt")

nltk.download("stopwords")

import pickle

tweet_filenames = sorted(glob.glob("C:\\\\Users\\ebiquity\\Downloads\\russiantweet.txt"))

print("Found files:")

tweet_filenames

corpus_raw = u""

for tweet_filename in tweet_filenames:

    print("Reading '{0}'...".format(tweet_filename))

    with codecs.open(tweet_filename, "r", "utf-8") as tweet_file:

        corpus_raw += tweet_file.read()

    print("Corpus is now {0} characters long".format(len(corpus_raw)))
```

```

    print()

tokenizer = nltk.data.load('tokenizers/punkt/russian.pickle')

raw_sentences = tokenizer.tokenize(corpus_raw)

def sentence_to_wordlist(raw):

    uni = b'\xd0\x9f\xd1\x80\xd0\xb8\xd0\xb2\xd0\xb5\xd1\x82,
    \xd0\xba\xd0\xb0\xd0\xba \xd0\xb4\xd0\xb5\xd0\xbb\xd0\xb0?'.decode('utf')

    print(re.findall(r'(?u)\w+', uni))

    clean = re.sub("r'(?u)\w+", " ", raw)

    words = clean.split()

    return words

sentences = []

for raw_sentence in raw_sentences:

    if len(raw_sentence) > 0:

        sentences.append(sentence_to_wordlist(raw_sentence))

print(raw_sentences[5])

print(sentence_to_wordlist(raw_sentences[5]))

token_count = sum([len(sentence) for sentence in sentences])

print("The russian tweet corpus contains {0:}, tokens".format(token_count))

num_features = 300

min_word_count = 3

num_workers = multiprocessing.cpu_count()

context_size = 10

downsampling = 1e-3

seed = 1

russian2vec = w2v.Word2Vec(

    sg=1,

    seed=seed,

    workers=num_workers,

    size=num_features,

    min_count=min_word_count,

    window=context_size,

    sample=downsampling

)

russian2vec.build_vocab(sentences)

```

```

print("Word2Vec vocabulary length:", len(russian2vec.wv.vocab))

russian2vec.train(sentences, total_examples=russian2vec.corpus_count, epochs=10)

if not os.path.exists("trained"):

    os.makedirs("trained")

russian2vec.save(os.path.join("trained", "russian2vec.w2v"))

russian2vec = w2v.Word2Vec.load(os.path.join("trained", "russian2vec.w2v"))

tsne = sklearn.manifold.TSNE(n_components=2, random_state=0)

all_word_vectors_matrix = russian2vec.wv.syn0

all_word_vectors_matrix_2d = tsne.fit_transform(all_word_vectors_matrix)


import numpy as np

import pandas as pd

import sys

import os

import re

import codecs

import csv

from keras.layers import Embedding

from keras.layers import Dense

from keras.layers import Input

from keras.preprocessing.text import Tokenizer

from keras.preprocessing.sequence import pad_sequences

from keras.models import Model

from keras.layers import LSTM

from keras.layers import Bidirectional

from keras import initializers

from keras import regularizers

from keras import constraints

from keras import optimizers

import keras.backend

import gensim.models.word2vec as w2v

cyber2vec = w2v.Word2Vec.load(os.path.join("Untitled Folder", "russian.word2vec.model")

```

```

cyber2vec.wv.save_word2vec_format("C:\\Users\\ebiquity\\Desktop\\russian.word2vec.txt",binary=False)

Vocab = 20000

Emb_D = 20

test_split = 0.2

batch_rate = 128

epochs = 100

samples = 10000

latent_dim = 256

source_data = []

target_data = []

target_data_offset = []

par_txt = 0

for line in open('C:\\Users\\ebiquity\\Documents\\rus.txt',encoding='utf-8'):

    par_txt += 1

    if par_txt > samples:

        break

    if '\\t' not in line:

        continue

    source_data, translation = line.rstrip().split('\\t')

    source_data.append(source_data)

    target_data.append(translation)

tokenize_data = Tokenizer(num_words=Vocab).fit_on_texts(source_data)

token2dim_source = tokenizer_inputs.word_index

print('Found %s unique input tokens.' % len(token2dim_source))

num_words_input = len(token2dim_source) + 1

tokenizer_outputs = Tokenizer(num_words=Vocab, filters='')

tokenizer_outputs.fit_on_texts(source_data)

target_sequences = tokenizer_outputs.texts_to_sequences(source_data)

word2dim_source = tokenizer_outputs.word_index

len_target = max(len(s) for s in target_sequences)

len_source = max(max_len_source, max_len_target)

output = len(token2dim_source) + 1

source_pad = pad_sequences(input, max_len=len_source)

targets_pad = pad_sequences(output, max_len=targets_pad)

```

```

num_words = min(Vocab, len(word2idx_inputs) + 1)
embedding_matrix = np.zeros((num_words, EMBEDDING_DIM))

for word, i in word2idx_inputs.items():
    if i < Vocab:
        embedding_vector = cyber2vec.wv[cyber2vec.wv.index2word[i]]

        if embedding_vector is not None:
            embedding_matrix[i] = embedding_vector

embedding_layer = Embedding(
    cyber2vec.wv.vocab,
    300,
    weights=[embedding_matrix],
    input_length=max_len_input,
)

target_one_hot = np.zeros(
    (
        len(source_pad),
        targets_pad,
        num_words_output
    )
)

for x, y in enumerate(targets_pad):
    for y, word in enumerate(x):
        target_one_hot[x, y, word] = 1

input_layer1 = Input(shape=(len_target))

emb = Embedding(
    um_words_input, Emb_D)

t = Bidirectional(LSTM(15, return_sequences=True))(t)

output = Dense

step_1 = Input(shape=(len(source_data),))

x = embedding_layer(step_1)

encoder = LSTM(
    latent_dim,
    return_state=True,
    dropout=0.5

```



```

)

encoder_outputs, h, c = encoder(x)

encoder_states = [h, c]

step_2 = Input(shape=(len(target_data),))

decoder_embedding = Embedding(num_words_output, latent_dim)

decoder_emb_train = decoder_embedding(step_2)

decoder_lstm = LSTM(

    latent_dim,

    return_sequences=True,

    return_state=True,

)

target_pad, _, _ = decoder_lstm(

    decoder_emb_train,

    initial_state=encoder_states

)

decoder_dense = Dense(num_words_output, activation='softmax')

target_pad = decoder_dense(target_pad)

model = Model([step_1, step_2], target_pad)


from nltk.corpus import wordnet

#load vocab file as a list -- english

words = [line.rstrip('\n') for line in open(english_vocab)]

syms = {w : [] for w in words}

for k, v in syms.items():

    for synset in wordnet.synsets(k):

        print synset.name.partition('.')[0]

        for lemma in synset.lemmas():

            v.append(lemma.name())

        print lemma.name.partition('.')[0]

print(syms)

with open(synset_english_xml, 'wb') as e:

    pickle.dump(syms, e)

```

## Bibliography

- [1] Ponnurangam Kumaraguru Aditi Gupta. Credibility ranking of tweets during high impact events. 2012.
- [2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.
- [3] Amir Bakarov. Improving word representations via global context and multiple word prototypes. 2018.
- [4] The U.S. Census Bureau. Internet world stats. <https://www.internetworldstats.com/>, Dec 2017.
- [5] R.Priyanga C.Sundar. Mining words and targets using alignment model. 2016.
- [6] DHS. Stop.think.connect. multilingual resources. <https://www.dhs.gov/stopthinkconnect-multilingual-resources/>, 2015.
- [7] Yoshua Bengio Dzmitry Bahdanau, Kyunghyun Cho. Neural machine translation by jointly learning to align and translate. 2014.
- [8] Jorg Tiedemann Elisabeth Niemann. The peoples web meets linguistic knowledge: Automatic sense alignment of wikipedia and wordnet. 2006.
- [9] Christian Girardi Emanuele Pianta, Luisa Bentivogli. Developing an aligned multilingual database. 2002.
- [10] Sebastien Jean Coline Devin Yoshua Bengio Felix Hill, Kyunghyun Cho. Embedding word similarity with neural machine translation. 2014.
- [11] Yuntian Deng Jean Senellart Alexander M. Rush Guillaume Klein, Yoon Kim. Opennmt: Open-source toolkit for neural machine translation. 2017.
- [12] Ludovic Denoyer Marc’Aurelio Ranzato Guillaume Lample, Alexis Conneau. Unsupervised machine translation using monolingual corpora only. 2017.
- [13] Ashok C. Popat Moshe Dubiner Jakob Uszkoreit, Jay M. Ponte. Large scale parallel document mining for machine translation. 2010.
- [14] Chengqing Zong Jiajun Zhang. Bridging neural machine translation and bilingual dictionaries. 2016.
- [15] Judith L. Klavans. Cybersecurity - whats language got to do with it? 2015.
- [16] Ray R. Larson. Introduction to information retrieval. 2009.

- [17] Jorg Tiedemann Lonneke van der Plas. Finding synonyms using automatic word alignment and measures of distributional similarity. 2006.
- [18] Claudio Schifanella Mario Cataldi, Luigi Di Caro. Emerging topic detection on twitter based on temporal and social terms evaluation. 2010.
- [19] T Mikolov and J Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.
- [20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *preprint arXiv:1301.3781*, 2013.
- [21] George A Miller. Wordnet: a lexical database for english. 1995.
- [22] Piyush Kedia Pushpak Bhattacharyya Mitesh M. Khapra, Sapan Shah. Projecting parameters for multilingual word sense disambiguation. 2009.
- [23] Sudip Mittal, Prajit Kumar Das, Varish Mulwad, Anupam Joshi, and Tim Finin. Cybertwitter: Using twitter to generate alerts for cybersecurity threats and vulnerabilities. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*, pages 860–867. IEEE, 2016.
- [24] Sudip Mittal, Anupam Joshi, and Tim Finin. Thinking, fast and slow: Combining vector spaces and knowledge graphs. *corpus*, 2:3, 2017.
- [25] Mu Li Ming Zhou Nan Yang, Shujie Liu and Nenghai Yu. Word alignment modeling with context dependent deep neural network. 2002.
- [26] NIST. Cybersecurity — nist. <https://www.nist.gov/topics/cybersecurity/>, 2018.
- [27] The Global WordNet Organization. Wordnets in the world. <http://globalwordnet.org/wordnets-in-the-world/>.
- [28] a Peter F. Brown, Jennifer C. Lai and Robert L. Mercer. Aligning sentences in parallel corpora. 1990.
- [29] The Register. Most vulnerabilities first blabbed about online or on the dark web. [https://www.theregister.co.uk/2017/06/08/vuln\\_disclosure\\_lag/](https://www.theregister.co.uk/2017/06/08/vuln_disclosure_lag/), Jun 2017.
- [30] The Register. Make america late again: Us 'lags' china in it security bug reporting. [https://www.theregister.co.uk/2017/10/20/us\\_china\\_vuln\\_reporting/](https://www.theregister.co.uk/2017/10/20/us_china_vuln_reporting/), Oct 2017.
- [31] Deepali Vora Suraj Subramanian. Unsupervised text classification and search using word embeddings on a self-organizing map. 2016.

- [32] Zareen Syed, Ankur Padia, M. Lisa Mathews, Tim Finin, and Anupam Joshi. UCO: A unified cybersecurity ontology. In *AAAI Workshop on Artificial Intelligence for Cyber Security*, pages 14–21. AAAI Press, 2015.

