POSTER: ALPACA-Advanced Linguistic Pattern and Concept Analysis Framework for Software Engineering Corpora

Phong Minh Vu,Tam The Nguyen,Tung Thanh Nguyen
Auburn University
lenniel@auburn.com,tam@auburn.com,tung@auburn.edu

Hung Viet Pham
University of Waterloo
hv.pham.2704@gmail.com

ABSTRACT

Software engineering corpora often contain domain-specific concepts and linguistic patterns. Popular text analysis tools are not specially designed to analyze such concepts and patterns. In this paper, we introduce ALPACA , a novel, customizable text analysis framework. The main purpose of ALPACA is to analyze topics and their trends in a text corpus. It allows users to define a topic with a few initial domain-specific keywords and expand it into a much larger set. Every single keyword can be expanded into long clauses to describe topics more precisely. ALPACA extracts those clauses by matching text with linguistic patterns, which are long sequences mixing both specific words and part-of-speech tags frequently appeared in the corpus. ALPACA can detect these patterns directly from pre-processed text We present one example demonstrates the use of ALPACA for text corpora of security reports.

KEYWORDS

Nature Language Processing, Linguistic Patterns, Trends

1 INTRODUCTION

In software engineering researches, domain specific corpora play a big part for information retrieval, topics, and trends analysis. However, unlike general text dataset, those dataset usually contain domain specific terms and expressions that are unique to their field (e.g. security domain with security terms, or mobile app reviews with fat-finger problem [8]). Using the current Natural Language Processing tools (NLP), such as StanfordNLP[3], LingPipe[1], or NLTK[2], researchers often have to work around with their own code to include domain specific vocabulary or even write their own preprocessor, which ultimately increases the amount of time needed to complete a study.

In some studies, tasks such as topic analysis, trend analysis [8, 9] were often done by grouping a set of words, or short phrases to describe the topic, then use them to search for the documents containing them to further study each topic. There has been no general approach for finding the clauses that explain topics of interest, despite of the more meaningful and context-rich nature of clause [9]. Inspired by those practical problems, we introduce ALPACA framework, a highly customizable domain specific text analyzer for: (1) Preprocessing text with customizable dictionaries; (2) Expanding topics from keywords and short phrases to clauses of topic details. (3) Analyze trends for expanded topics. Moreover, ALPACA also offers the following valuable artifacts: A re-purposed English dictionary from WordNet for root word finding and POS tagging; A growing list vocabulary of IT, technical and mobile app terms collected from previous studies, re-purposed for root word finding and POS tagging; A growing list of domain specific linguistic patterns.

ICSE '18 Companion, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5663-3/18/05.
https://doi.org/10.1145/3183440.3194972

timecard cms allow remote attacker to execute arbitrary sql command
execute arbitrary sql command via the execute query array
myphpnuke allow remote attacker to execute arbitrary php code

Table 1: Examples of clauses extracted for SQL Injection

2 RUNNING SCENARIO: CVE TRENDING

Nehaus et al published a trend analysis[6] for security problems in Common Vulnerability and Exposures (CVE) dataset in 2009. However, the security trends may have changed for the past 6 years. Therefore, we picked one major topic (SQL Injection) from their study, and used ALPACA on up-to-date CVE data to answer a research questions: How did the trends changed after 2009?

The data we used consists of 112,197 CVE reports from the CVE database, dated from 1999 to the end of 2016. On this dataset, we used ALPACA to preprocess the text to root words level to minimize the semantic that can be lost using Snowball Stemmer[7] over-stemming method [8]. This pre-processed text will then also be combined with the English Wikipedia dump corpus for Word2vec training [4]. Since Word2vec's performance depends on the size of the corpus, this combination is necessary for CVE dataset, which only contain very short and descriptive documents. The approach reinforces the context vector of common words that appear in both corpora, while the technical and unique words of CVE dataset still have the same context vector.

The results of ALPACA preprocessing also include a few predefined scores computed for each word. These scores include IDF, frequency and Weibull score on frequency. For the CVE dataset, we assumed the most general but technical and descriptive words follow a Weibull distribution, therefore we picked Weibull as the scoring scheme for our words.

Next, we fed ALPACA with the topic keywords for SQL Injection, Cross-site Scripting, and Buffer Overflow discovered by Nehaus et al . The results are shown in Table 1 and Figure 1. As shown in Figure 1, even though ALPACA used a different approach to Nehaus's, the shape of the trends from 2000 to 2009 is similar to their findings. In the years after 2009, it had a sharp decline in the percentage of

Figure 1: **Trends analysis of CVE dataset**

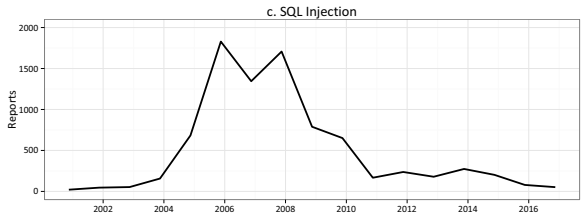
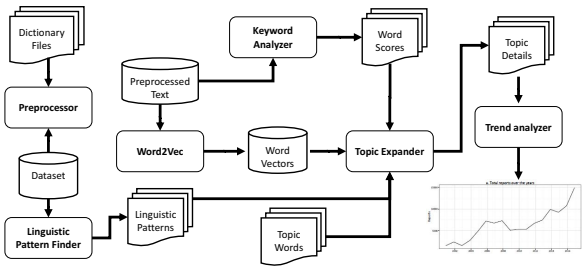


Figure 2: **Overview of ALPACA**



this type of attack during the last few years. This could indicate that the SQL Injection is not a popular breach anymore.

3 TECHNICAL APPROACH

Figure 2 shows the overall of ALPACA framework. The input for each task module can be modified outside of the framework and will directly affect the results. Modifiable input includes: Dictionary files, Linguistic patterns, word scores, topic words, topic details.

3.1 Preprocessing

To address a wide variety of needs for text analysis tasks, we had designed ALPACA to be highly customizable. First, the preprocessing module can be set to one of the three levels: LV1-word correction using a custom spelling corrector (for tasks such as vocabulary discovery); LV2-Root word stemming with predefined dictionaries (for tasks such as trends discovery, topic expansion and opinion mining); LV3-over-stemming with Snowball Stemmer (for tasks such as searching and information retrieval).

The predefined dictionaries are provided by the users for flexibility in adding domain specific vocabulary. Along with the tool, we also provided a default dictionary of English words extracted from WordNet 3.0 [5], a domain specific vocabulary for reviews from previous studies [10], Linux dictionary, and a list of functional words (connectors, intensifiers, negations, wh [9]). We will continue adding more domain specific vocabularies in the future.

3.2 Pattern extraction and matching

Originally suggested by Vu et al 's idea of extracting phrase template[9] for faster mapping of phrases, we expanded the concept of phrases into more informative clauses patterns. The patterns are originally extracted from Stanford phrasal extraction module [3], then simplified into a structure of Noun phrases, Verb phrases and Adjective phrases all held together by functional words. The intuitive is that functional words often carry no information on their own, but instead serve as a binder between smaller structures to make a more informative sequence such as the ones shown in Table 1. Users can also add more functional words for their domain of interest, enabling the capturing of heuristic patterns for specific domains.

To match the patterns with actual text, on each sentence, ALPACA starts with a list of "seeds", which are the homogeneous Noun/Verb/Adjective phrases (i.e. phrase containing only one kind of POS tag) that contains no functional word. This list is then ranked by a structural scoring scheme and the highest ranking seed will be merged to a neighboring seed to create a higher scoring and bigger seed. The merging process add all the functional words in between the seeds. The algorithm then computes the rankings again and repeat the merging process until there is no seed left to merge or merging will not create a higher scored seed. This algorithm tries to find the local optimal results of the ranking scheme. Our structural scoring scheme, shown in Formula 1, aims to favor the sequence with more meaningful word but not too long and has as many functional words as possible. The reason is based on our observation that structural words bind original seeds together to form more complex and meaningful sequence. Word's weights are for usersto choose from our default weights (IDF, frequency, Weibull, rating contrast score [8], or mean values) or add their own.

$P : (W : \mid k :)$

$k \in P$: number of functional words

$W \in P : (w_1..w_n)$ A list of non-functional words

$$score(P) = \frac{klog(n)\sum_1^n w_i}{n} \quad (1)$$

3.3 Trend Analysis

For analyzing trends of a topic, ALPACA uses the expanded statements as input for finding documents that contains such topic and count their statistics. We provided 3 default approaches to find related documents: 1-matching any document that has exact matches of keywords and clauses in the expanded topic. 2-Similar to the first approach, but only matches the clauses. 3-Compute portion of 1-gram, 2-grams and 3-grams that matched with the topic clauses and compare them to a predefined threshold.

REFERENCES

[1] Breck Baldwin and Bob Carpenter. [n. d.]. LingPipe. ([n. d.]).
[2] Steven Bird. [n. d.]. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*.
[3] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. [n. d.]. Generating typed dependency parses from phrase structure parses. In *LREC*.
[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. [n. d.]. Efficient Estimation of Word Representations in Vector Space. *CoRR* ([n. d.]).
[5] George Miller and Christiane Fellbaum. 1998. Wordnet: An electronic lexical database. (1998).
[6] Stephan Neuhaus and Thomas Zimmermann. [n. d.]. Security trend analysis with cve topic models. In *ISSRE'10*.
[7] MF Porter and Richard Boulton. 2001. Snowball stemmer. (2001).
[8] Phong Minh Vu, Tam The Nguyen, Hung Viet Pham, and Tung Thanh Nguyen. [n. d.]. Mining User Opinions in Mobile App Reviews: A Keyword-Based Approach (T). In *ASE'15*.
[9] Phong Minh Vu, Hung Viet Pham, Tam The Nguyen, and Tung Thanh Nguyen. [n. d.]. Phrase-based Extraction of User Opinions in Mobile App Reviews. In *ASE'16*.
[10] Phong Minh Vu, Hung Viet Pham, Tam The Nguyen, and Tung Thanh Nguyen. [n. d.]. Tool Support for Analyzing Mobile App Reviews. In *ASE'15*.