2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

IKEA: Unsupervised domain-specific keyword-expansion

Joobin Gharibshah, Jakapun Tachaiya, Arman Irani, Evangelos E. Papalexakis and Michalis Faloutsos
University of California - Riverside, CA
Email: {jghar002,jtach001,airan002,epapalex,michalis}@cs.ucr.edu

***Abstract*—How can we expand an initial set of keywords with a target domain in mind? A possible application is to use the expanded set of words to search for specific information within the domain of interest. Here, we focus on online forums and specifically security forums. We propose IKEA, an iterative embedding-based approach to expand a set of keywords with a domain in mind. The novelty of our approach is three-fold: (a) we use two similarity expansions in the word-word and post-post spaces, (b) we use an iterative approach in each of these expansions, and (c) we provide a flexible ranking of the identified words to meet the user needs. We evaluate our method with data from three security forums that span five years of activity and the widely-used Fire benchmark. IKEA outperforms previous solutions by identifying more relevant keywords: it exhibits more than 0.82 MAP and 0.85 NDCG in a wide range of initial keyword sets. We see our approach as an essential building block in developing methods for harnessing the wealth of information available in online forums.**

INTRODUCTION

What are the relevant keywords to search a forum so that we can maximize the amount of useful information that we can extract? This is the overarching question that motivates this work. First, we argue that there is a wealth of information in online forums. These forums aggregate the collective wisdom of millions of people around the world, and they capture useful information, signals and trends. Second, we focus on security forums to ground our work. Thus, our goal is to help a security analyst by making our approach: (a) easy to use, by asking few initial keywords, and (b) flexible to cater to a wide range of types of investigations.

We define the problem in more detail. The user provides: (a) an initial set of keywords, (b) a sample forum, and (c) her expansion preference. The output is an expanded set of **ranked** keywords from the sample forum that best relates to the initial keywords. We consider the following requirements. First, we want our approach to work even with a really small set of initial keywords. Second, we want to enable the user to get the answer that best matches their intention, which we explain below.

We provide an example to clarify the problem and the concept of user preference. The user could provide two keywords *virus* and *attack* or a name of a virus. The goal is to retrieve the most relevant and important keywords leveraging the sample forum. We consider this as the **default** type of user intention, where a response could include words like *malware, ransomware*, or *antivirus*. However, a user may have a preference towards identifying specific names of malware, tools or technical jargon, in which case they would prefer an answer that would include words like *kraken* and *rustock*. Namely, the goal is to return the proper names and jargon, akin to the system learning by example in an unsupervised fashion. We use the term **jargon-focused** for this type of user preference.

The above problem formulation has received relatively little attention, and usually in a tangential way. We can group prior work as follows. First, there have been several embedding-based techniques for query expansion [1], [2], [3], which we compare with our approach in sections *Experimental Results* and *Related Work*. Second, some efforts focus on topics and keyword extraction from a document [4], [5], without an initial keyword set. Third, other studies apply NLP-based techniques to identify specific information and user interactions, such as malicious IP addresses, and selling of services in security forums [6], [7], [8]. We elaborate on previous work in section *Related Work*.

We propose a systematic approach, IKEA[1], to expand an initial limited set of keywords focusing on a specific domain in an unsupervised learning fashion. The novelty of our approach is three-fold: (a) we use and combine two similarity expansions in the word-word and post-post spaces, in an appropriately constructed embedding space, (b) we use an iterative approach for each of the aforementioned expansions, and (c) we provide a flexible processing of the identified words. The flexibility in the last step refers to our ability to rank the retrieved words in the order that best suits the needs of the user query as in the example mentioned above [2].

We provide a high-level view of our approach in Figure 1. It shows how we start from an initial set of keywords, which we project in appropriately-defined embedding space. Second, we use an iterative process to identify similar keywords, set K_s, in the word embedding space (Panel 2). Third, we identify the posts, P_i, that contain keywords from the set K_s. Fourth, we use an iterative process to identify a set of similar posts,

[1]IKEA stands for **I**terative **K**eyword **E**xpansion **A**pproach.

[2]The jargon-focused case can be seen as a variation of the named entity identification. Here, we address a *targeted* named-entity identification, a less-explored variation, and our goal is to showcase the flexibility introduced by the last step in our approach.

IEEE/ACM ASONAM 2022, November 10-13, 2022
978-1-6654-5661-6/22/$31.00 2022 IEEE

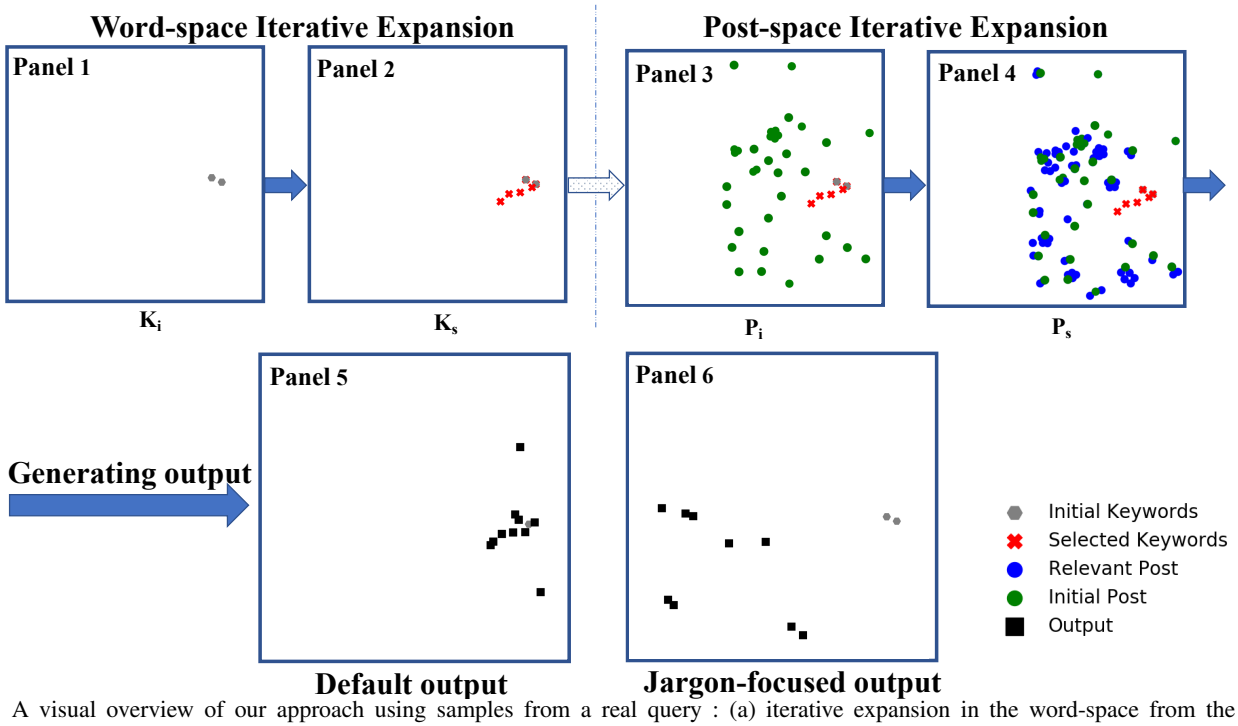

Fig. 1: A visual overview of our approach using samples from a real query : (a) iterative expansion in the word-space from the initial keywords (Panel 1, gray circles) to the expanded keyword set K_s (Panel 2, red crosses); (b) iterative expansion in the post-space from the posts P_i obtained using the K_s keywords (Panel 3, green circles) to the extended post set P_s (Panel 4, blue circles). Output: the top 10 highest ranked words based on user preference for default in Panel 5, and jargon-focused in Panel 6.

P_s, in the post embedding space (Panel 4). Finally, we extract keywords from the set of posts P_s and present them to the user ranked according to their preference.

We evaluate our method using three security forums over a five-year period. For the evaluation, we created a labeled dataset using both: (a) the Mechanical Turk service, and (b) security experts. We intend to make available our annotated dataset to the community in order to facilitate further research in this space.

We summarize our key results below:

- **IKEA outperforms other state-of-the-art methods**. Specifically, IKEA exhibits more than 0.82 Mean Average Precision (**MAP**) and 0.85 Normalized Discounted Cumulative Gain (**NDCG**) in the top 50 retrieved keywords on average across three forums.
- **IKEA finds relevant jargon-focused keywords with up to 0.94 precision.** The flexible ranking empowers IKEA to exhibit relatively good precision. Interestingly, we find that 35% of these keywords are names of malware and virus, as we see in section *Experimental Results*.
- **IKEA works well as a query expansion method for documents.** Stepping away from forums, we use IKEA as a query expansion technique on the Fire 2011 document-query benchmark. We find that IKEA outperforms other state-of-the-art methods with a MAP of 0.33-0.41.

The overarching vision is to provide a powerful, easy-to-use, and flexible method to provide domain-specific keyword expansion in an unsupervised way. We see our approach as a key capability within a practical tool-set for harnessing the wealth of information in online forums.

Background and Datasets

Our work focuses on security forums, but we also consider a document-based benchmark. We discuss our datasets below, and present their basic statistics in Table I.

1. Security Forums. We have collected data from three different forums: OffensiveCommunity (*OC*), HackThisSite (*HT*) and EthicalHackers (*EH*). These forums bring together a wide range of users: system administrators, white-hat hackers, black-hat hackers, and users with variable skills, goals and intentions.

We briefly describe our three forums below.

a. OffensiveCommunity (OC): As the name suggests, this forum contains “offensive security” related threads, namely, breaking into systems. Many posts consist of step by step instructions on how to compromise systems, and advertise hacking tools and services.

b. HackThisSite (HTS): As the name suggests, this forum has also an attacking orientation. There are threads that explain how to break into websites and systems, but there are also more general discussions on cyber-security.

c. EthicalHackers (EH): This forum seems to consist mostly of “white-hat" hackers, as its name suggests. However, there are many threads with malicious intentions in this forum.

2. Document benchmark: Fire 2011 (English). This is an annotated benchmark dataset for information retrieval pur-

TABLE I: The basic statistics of our forums and Fire 2011 dataset. Fire consists of documents instead of posts.

	OffensComm.	HackThisSite	EthicalHackers	Fire
Posts	25,538	84,745	54,176	89,286
Threads	3,542	8,504	8,745	N.A
Words	45,119	47,810	48,157	551,075

TABLE II: The list of our initial keyword sets.

Category	Identifier	List of keywords
Security	W_{MV}	Malware, Virus
	W_{HA}	Hack, Account
	W_{AV}	Attack, Vulnerability
Financial	W_{CC}	Credit, Card, Bank
	W_{SB}	Buy, Sell
Video Tutorial	W_{VG}	Video Game
	W_{TG}	Tutorial Guide
Jargon	J_{OC}	Darkcomet, Gingerbread
	J_{HT}	Morris, Slowloris
	J_{EH}	Chernobyl

TABLE IV: The symbol table with the key notations.

Symbol	Description
K_i	Initial set of keywords
K_s	Selected set of keywords in the word-based iterative process
K_f	Keywords appearing in a forum
P_i	Initial set of posts containing K_s
P_s	Selected set of posts after the iteration
P_f	Posts appearing in a forum
Z_k	Keywords similarity threshold
Z_p	Posts similarity threshold
K_e	Keywords extracted from P_s
Sim_K	Keywords similarity function
Sim_P	Posts similarity function
α_i	Ranking score weight for parameter i

poses. It consists of documents from an English news agency, and 51 queries with the relevant documents.

Initial keywords: We evaluate the performance of our approach on the forums dataset with initial indicative sets of keywords as shown in Table II. In practice, the keywords will be determined by the interests of the person using our approach, such as a security analyst. To ensure breadth, we use keyword sets that relate to different categories of queries as shown in Table II. To stress test our approach, we focus on keyword sets with less than three words, which arguable makes the life of the user easier. Note, that we did experiment with three or more keywords, and the results were qualitatively similar with our approach performing well.

TABLE III: Assessing the annotator agreement using the Fleiss-Kappa coefficient for each initial keyword set experiment.

Identifier	MTurk	Experts
W_{AV}	-	0.569
W_{HA}	-	0.535
W_{MV}	0.436	0.652
W_{CC}	0.444	0.511
W_{VG}	0.699	-
W_{TG}	0.677	-
W_{SB}	0.672	-
J_{Jargon}	-	0.626

Establishing the groundtruth. Despite some recent efforts [6], [9], we were not able to find any benchmarks for online forums.

To establish the groundtruth, we use two group of annotators to evaluate the relevancy: (a) five experts in the security domain, and (b) five annotators from Amazon's Mechanical Turk platform (www.mturk.com). The annotators labeled the keywords based on the relevancy to the initial keyword set. In more detail, each word is labelled as relevant (*"a synonym, or a potential companion of the initial keywords in an English technical text"*). The final label is produced by using the majority vote approach. As expected, the Mechanical Turk annotations were of poor quality on the security related keyword sets as we discuss below. This happened despite setting high criteria to get only skilled annotators.

We also need groundtruth for assessing the ability of our approach in finding jargon-focused keywords. Here, we use security experts to label the retrieved keywords, since the subject and the context in this type of queries require even more technical expertise. In more detail, we provide our experts with the following context: (a) the keyword, (b) a post snippet that contains the keyword, and (c) top-ten google search results on the given keyword.

We assess our annotated data by using the Fleiss-Kappa coefficient on the two groups of annotators and we show the coefficient on average across three forums in Table III. We see that there is good agreement as the Fleiss-Kappa coefficient is in the range of 0.677 and 0.699. Two queries, W_{MV} and W_{CC} , have been labeled by both groups of annotators and the expert annotators show the higher inter-agreement coefficient. We see a coefficient of 0.652 for experts versus 0.436 for MTurks in the case of W_{MV}. This suggests the need to use experts as the annotation tasks become more technical.

OVERVIEW OF IKEA

Our approach provides a domain-specific keyword expansion consisting of four major steps, which we outline below.

Step 1: Domain representation. We represent words and posts of forums in an m-dimensional embedding space.

Step 2: Word-space expansion. We expand the initial set of keywords by adding relevant words iteratively.

Step 3: Post-space expansion. We identify posts that are similar to the set of posts, which contain the relevant words from the previous step.

Step 4: Result Processing. We extract and rank the keywords from the posts of the previous step, based on several metrics of importance and relevancy.

In the rest of the section, we discuss algorithmic aspects of the above steps, and we highlight their novelty. Note that an additional novelty is the combination of all these elements in an effective framework.

Step 1: Domain representation

We project words and posts in an appropriately-constructed m-dimensional space. Here, we use the Word2Vec approach [10] as a building block for doing this projection. We project every post on the same m-dimensional space by using the average of the projections of its words. There are other methods to project posts in an embedding space [11], [12], which we will evaluate in the future. Our current approach gives sufficiently good results.

Step 2 and 3: Two iterative expansions

We propose an iterative approach for establishing similarity between: (a) words, and (b) posts. This is part of our novelty, since, to the best of our knowledge, no prior work used such iterative approaches within an embedding representation.

a. Word-space Iterative Expansion. We expand the initial keyword K_i into set K_s adding similar keywords iteratively. In each step, we include words whose average similarity to any of the words in K_s is above a **threshold** Z_k. This threshold takes values in the range [0-1], with lower values leading to more selected words. We repeat this process, until we cannot identify any more words for inclusion. In Figure 1, we depict this expansion as the transition between panel 1 and panel 2.

b. Post-space Iterative Expansion. As mentioned earlier, we identify the posts, P_i, that contain keywords from the set K_s. We then apply a similar iterative process to expand P_i into the P_s set of posts. In each step, we add more relevant posts to P_s and we stop, when no more posts can be added using the same threshold as above. This process is represented by Panel 3 and Panel 4 in Figure 1.

Why an iterative approach makes sense: In Figure 2, we show the intuition behind our choice of an iterative approach. We depict a word in the initial set with a "black star". and find relevant words shown with "black diamonds". The iterative process leads to a chain-like selection of similar keywords. This way, the selected words are "very" close to either the initial set of keywords, or words that were previously selected as similar.

Could we achieve the same by having a lower similarity threshold Z_k? This would be equivalent to enlarging the radius of our "similarity" circle as shown in Figure 2. However, if we did that, we would run into the risk of including words that are typically far away from any of the initial or selected words.

This intuition is corroborated by our experiments. We vary the similarity threshold, Z_k and observe its effect on the quality of the keywords retrieved with the word-space iterative approach (focusing on the top 20 words) in Figure 3. We see that, by reducing the threshold Z_k from 0.95 to 0.8, we get 45% more irrelevant keywords in the word-space iterative expansion.

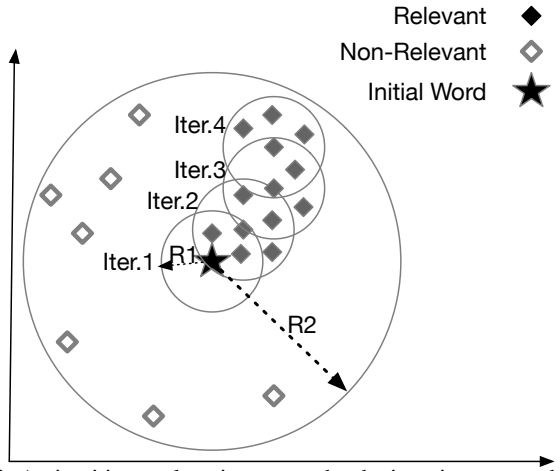

Fig. 2: An intuitive explanation as to why the iterative approach gives better results.

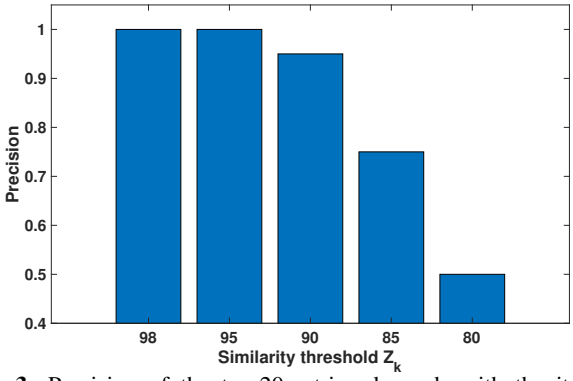

Fig. 3: Precision of the top-20 retrieved words with the iterative approach in IKEA for different Z_k values.

Step 4: Result Processing

Our approach introduces a processing stage to further refine the results in order to better respond to the user's needs. This is achieved with two main capabilities, which we describe below.

(a) Filtering words: We have the ability to do an optional filtering on the words based on the user's needs. We can remove the keywords that appear in a dictionary or blacklists provided by the user. This provides the ability to remove words that the user knows are not of interest.

(b) Ranking words: An additional functionality is to rank the extracted keywords, in the order that is more likely to be of interest to the user. There are many different types of questions that the user may be trying to answer, and there are also various ways to rank words. As a proof of concept, we provide currently two options for ranking, as we discussed in the example in the introduction: a) **default**: where we rank words in terms of both popularity and their similarity to the words of interest, b) **jargon-focused:** where we prioritize "jargon" words, such as names of malware, antivirus, and technical terms etc. We provide more details for our ranking below.

A. Metrics of importance and relevance. We use the following metrics to quantify aspects of the relevance, uniqueness and frequency of the words:

1. Word-Word similarity, $Sim_K(w, K_i)$: This functions captures the similarity of word w with the initial set of words provided by the user, K_i. We use the average cosine similarity between all words in K_i, which is a widely used metric in this space [1], [2].

2. Word-Post similarity, $Sim_P(w, P_s)$: This is a recently introduced metric, which captures the relevance and significance of keyword w to the posts that it appears [4]. Here, we use the metric to capture the "closeness" of word w to the set of posts P_s by calculating the average cosine similarity between word w and each post.

3. TFIDF, $TFIDF(w)$: Term Frequency-Inverse Document Frequency is a widely-used metric in information retrieval, which captures how important a word, w, is to a document within a collection of documents [13]. The intuition is that if a word is relatively rare overall, its appearance in a post is more significant compared to a word that appears in every post.

4. Inverse Document Frequency, $IDF(w)$: Inverse Document Frequency, $IDF(w)$ shows the reciprocal of frequency of posts containing word w in a corpus of documents, (the P_s set in our case). This metric measures the rarity of the word, and hence its discerning power and is widely used in this space [13].

Note that, TFIDF and IDF are metrics that capture the discerning power of a word given a set of documents/posts in a complementary way. IDF is particularly useful in the jargon-focused case.

B. Word ranking function. To rank the words, we combine the above metrics using a weighted function as follows in the default mode:

$$R_{Def}(w) = \alpha_1 * Sim_K(w, K_i) + \alpha_2 * Sim_P(w, P_s) + \\ + \alpha_3 * TFIDF(w) + \alpha_4 * IDF(w) \quad (1)$$

where $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$.

There are different ways to assign weights. Here, we use the rank exponent weight method [14], in which the normalized rank of each metric defines its weight. The weight α_{r_i} of rank r_i is given by:

$$\alpha_{r_i} = \frac{(N - r_i + 1)^\rho}{\sum_{k=1}^{N}(N - r_k + 1)^\rho} \quad (2)$$

where N=4. We find the best performance for ρ=2 experimentally. Intuitively, as we increase the value of ρ, we decrease the weight of the weight of the lower-ranked metrics. As we will see later, this approach seems to work well.

In the jargon-focused case (as defined earlier), we use the same function, but we change the order, and hence the α_i weights for each metric:

$$R_J(w) = \alpha_1 * IDF(w) + \alpha_2 * TFIDF(w) + \\ + \alpha_3 * Sim_K(w, K_i) + \alpha_4 * Sim_P(w, P_s) \quad (3)$$

C. User preference: We enable the users to specify the preferred order of the keywords. We currently provide two options to the user:

a. Default preference: We use no filtering, and the ranking order in Eq. 1, which gives more weight to the similarity to the initial keywords.

b. Jargon-focused preference: We filter out English words, and use the ranking order in Eq. 3, to prioritize unusual words with highly discriminative power, which points us to jargon words.

D. Advanced user customization options. We have identified several opportunities to customize our approach. A sophisticated user can: (a) provide blacklists and dictionaries in the filtering stage, (b) introduce different ranking weights and functions. These are things that we will explore in the future as we discuss in section *Discussion*.

EXPERIMENTAL RESULTS

In this section, we present the evaluation of our approach.

Data and queries. We use three security forums and 10 queries as we described in section *Background and Datasets*. When needed, we use a publicly available english dictionary (github.com/dwyl/english-words).

Defining our embedding space. We use a well-established skip-gram Word2Vec embedding approach [10]. First, we remove stop words, URLs and html tags and use Porter Stemmer for the stemming of words. In our Word2Vec model, we set the dimension of the embedding space for words and posts to 100. Experimentally, we opt for a value of 5 for the *training window size* parameter, which determines how much context around a word we consider during the training of the model [10]. In the iterative approach, we set the similarity threshold Z_k to 0.90.

Evaluation Metrics. We use the following metrics for evaluation: (a) precision, defined as the probability that an identified word is indeed relevant, (b) the mean average precision (MAP), when aggregating over multiple queries, and (c) the normalize discounted cumulative gain (NDCG). NDCG is a widely-used metric, which quantifies the quality of a ranking. We use the commonly-used paired t-test to evaluate the significance mean difference of the results [3], [1].

Reference methods. We evaluate our approach against other state-of-the-art methods for keyword expansion. We briefly describe them below, and we provide a detailed discussion and comparison with our approach in section .

- **Query expansion with maximum likelihood estimate(QM):** [2] This method uses Word2Vec embedding to find similar words to a given query in the word space language model.
- **Automatic Query Expansion (AQE):** [3] This is a a query expansion technique, where related words are ranked using the K-nearest neighbor approach.
- **Iterative thesaurus-based approach:** This is a straightforward method, which returns synonyms of the initial keywords using a thesaurus capability of a dictionary (e.g. http://thesaurus.com). We obtain a list of synonyms to the

TABLE V: Default: Comparison of performance with mean average precision (MAP) and normalized discounted cumulative gain (NDCG). We use bold for the cases where IKEA exhibits the highest performance, and use the "cross"(†) when a reference method matches that performance. The asterisk (∗) indicates the significance statistics using paired t-test with 95% confidence interval measure.

	IKEA		AQE		QM		
	MAP	NDCG	MAP	NDCG	MAP	NDCG	Datasest
@10	**0.957**	**0.968***	0.957†	0.951	0.957†	0.922	
@20	**0.957***	**0.947***	0.871	0.926	0.843	0.896	
@30	**0.895***	**0.922***	0.833	0.899	0.767	0.889	OffensiveCommunity
@40	**0.871***	**0.905**	0.811	0.903	0.746	0.879	
@50	**0.829***	0.891	0.803	**0.904**	0.714	0.882	
@10	**1***	**0.983***	0.971	0.965	0.957	0.895	
@20	**0.957***	**0.937***	0.929	0.924	0.9	0.902	
@30	**0.938**	**0.920**	0.928	0.918	0.866	0.901	HackThisSite
@40	**0.9**	0.893	0.89	**0.898**	0.818	0.876	
@50	**0.9**	**0.892**	0.90†	0.892†	0.814	0.883	
@10	**0.986***	**0.970***	0.971	0.951	0.571	0.642	
@20	**0.9***	**0.921***	0.871	0.910	0.671	0.688	
@30	0.857	**0.885**	**0.861**	0.885†	0.695	0.699	EthicalHackers
@40	**0.839**	**0.869**	0.839†	0.861	0.703	0.705	
@50	**0.826***	**0.858***	0.808	0.841	0.694	0.705	

initial keywords and expand them, similar to our iterative approach, until there are no new words added.

Default user preference

We show that IKEA outperforms the reference methods. In Table V, we compare the three approaches using two metrics (MAP and NDCG) in our three forums. In our comparison, we vary the number of the top retrieved keywords: @10 means the top-10 keywords. In Table VI, we show the top-10 expanded keywords for the queries introduced in Table II.

a. MAP: IKEA outperforms the competition in 14 out of 15 cases. As shown in Table V, IKEA performs as well or better in the majority of the cases compared to the other methods with respect to MAP. We highlight (with a asterisk ∗) the cases where the paired t-test indicates statistically significant performance difference.

IKEA exhibits good performance: MAP is more than 0.826 across all the different experiments. In other words, IKEA returns a relevant word 82.6% of the time. We see that IKEA significantly outperforms the other two methods in the case of the top 20, 30, 40, and 50 retrieved keywords.

b. NDCG: IKEA outperforms the competition in 13 out 15 cases. Focusing on NDCG, IKEA again performs at least as well as the competition in 13 experiments. Furthermore, if we focus on the top-10 and top-20 extracted keywords, IKEA is better than the others method among all forums. Recall that NDCG is an indication of the ranking quality: a high NDCG value indicates superior ranking quality with the most relevant words near the top.

c. Thesaurus-based approach identifies only 16% words reported by IKEA. A natural question is: Does IKEA add more value than a simple thesaurus search? The answer is yes. In Figure 4, we plot the percentage of common keywords retrieved by IKEA, AQE and QM compared to the thesaurus-based approach for the top 100 words. The thesaurus-based approach finds at most 16% of the keywords retrieved by IKEA and AQE, and much less (≤6%) by QM.

We attribute the difference to the fact that a thesaurus-based approach is not domain-specific, but relies on word similarity broadly-defined at the dictionary level.

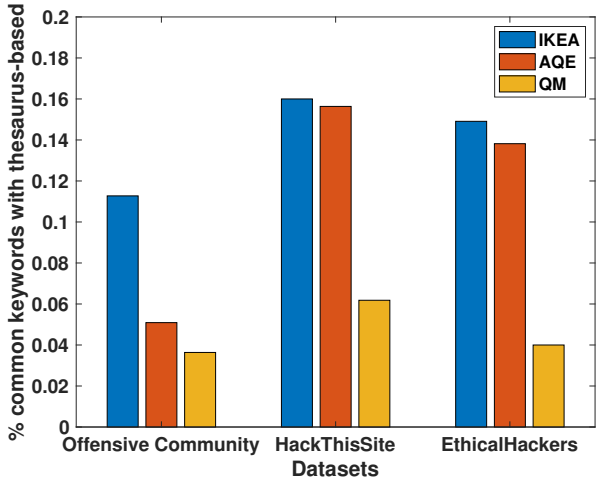

Fig. 4: The percentage of common words between the retrieved keyword from the three methods and the thesaurus-based approach in our forums.

Jargon-focused user preference

We evaluate IKEA in the jargon-focused case.

IKEA finds jargon words with up to 0.94 precision. We show the performance of IKEA in the jargon-focused case in Table VII for the top 50 and top 100 keywords. Further, we evaluate our combined four-metric ranking compared to using only one metric. We consider two alternative ranking functions using the first metric (with the most weight) from Eq. 1 and 3: (a) $R_{Sim}(w) = Sim_K(w, K_i)$, and (b) $R_{Freq}(w) = IDF(w)$. We see that our combination of the metrics, gives better results compared to single metric rankings.

TABLE VI: The expanded keyword sets with IKEA for our initial keyword sets. The underlined words are virus or malware names.

Initial	Retrieved top-10 ranked keywords
W_{MV}	malware, virus, trojan, infect, antivirus, malwarebytes, rootkit, worm, avg, spyware, investigate, keylogger
W_{HA}	hack, account, hotmail, facebook, twitter, gmail, bank, banking, learn, teach, instagram
W_{AV}	attack, vulnerability, phish, deface, defacement, ddos, dos, ftp, victim, credential, gmail, steal
W_{CC}	credit, card, bank, rfid, account, driver, wireless, sim, transfer, deposit, cash, subscription
W_{SB}	sell, buy, pay, cheaper, exchange, cash, tax, earn, purchase, reputation, fee, paypal
W_{VG}	video, game, youtube, play, vid, music, movies, watch, rpg, clip, mmorpg, fun
W_{TG}	tutorial, guide, tuts, beginner, noobie, teach, mentor, tip, help, recommend, explain, advice
J_{OC}	darkcomet, gingerbread, smp, hideman, cwm, msfconsole, adb, battlefield, casperspy, uniscan, urllib, fadias
J_{HT}	morris, slowloris, ugand, revolutinaryg, imf, lov, knoppix, openvms, teabag, joli, virtuawin
J_{EH}	chernobyl, zeroaccess, athcon, mersenne, duronio, dubuque, crypter, rustock, maricopa, wua, pornography, Gaobot

TABLE VII: Jargon-focused: Precision of the top 50 and 100 ranked keywords.

	IKEA	R-Sim	R-Freq	Forum
@50	**0.62**	0.56	0.4	OffensiveCommunity
@100	**0.584**	0.495	0.426	
@50	**0.50**	0.45	0.2	HackThisSite
@100	**0.41**	0.38	0.36	
@50	**0.941**	0.902	0.902	EthicalHackers
@100	**0.901**	0.851	0.871	

Identifying emerging malware is a key concern in the security world, while jargon can include other technical terms. Upon manual investigation, we find that 35% of the retrieved words with our approach are indeed malware and virus names.

IKEA: query expansion on the Fire dataset

As a case study, we evaluate our method within the context of informational retrieval focusing on the following problem. Given a set of documents and a query in natural language (NL), we want to find the related documents to that query. We use IKEA as a building block: (a) we extract keywords from the query, (b) we use IKEA, and (c) we used Lucene search engine for document retrieval and rank them accordingly [3]. To compare, we repeat the process replacing IKEA with the other two methods, AQM and QE, in the second step.

IKEA outperforms or matches the other state-of-the-art methods as a building block in the document retrieval framework outlined above. We apply the three approaches to the set of queries in the Fire 2011 dataset, which we described in section *Background and Datasets*.

In Table VIII, we show the average over 10 sample queries in Fire 2011 using both precision (MAP) and ranking quality (NDCG). This results show the impact of IKEA compare to no expansion and other embedding approaches without proposed iterative expansion in IKEA. We see upto 39.5% improvement in NDCG and upto 43.9% improvement in MAP. This values show significant improvement in the performance while we use IKEA as the keyword expansion approach.

Related Work

We summarize related work in the following general areas.

a. Embedding approaches in query expansion. Several recent efforts leverage embedding approaches in extending a query, usually for structured documents, such as news reports [1], [15], [3].

We discuss the two most relevant studies and compare them with our approach. The QM approach [1], [2], [16] uses embedding to identify similar words in the word-space only. The AQE approach [3], [17] uses similarity expansion in the word-space selected from sudo relevant documents and they use a nearest neighbor technique to rank keywords. They differ from our work in that they do: (a) not use an iterative expansion in word and post domain (QM and AQE), (b) not do a similarity expansion in the post-space (QM), (c) not have a flexible ranking capability (QM and AQE). We argue that combination of all the above elements contributes to the superior performance and flexibility of our approach.

Following a different path, some methods rely on user-behavior and feedback to establish a statistical model of the word relevancy [18], [15].

b. No initial keyword set: extracting topics and keywords. Though related, these works address a fairly different problem than we do here: the goal is to identify the "important" words in a document without an initial keyword set. Several works identify keywords or key-phrases that capture the topic of a document [4], [19], [20].

Named-entity extraction is a tangentially related problem to our jargon-focused case with several supervised [21], [22] and unsupervised approaches [23], [24], which usually do not assume an initial keyword set.

c. NLP-based techniques for extracting specific information. A group of recent studies focus on retrieving specific information, such as: (a) prices and availability of malicious services in security forums [7], [25]; (b) extracting malicious IP addresses and discussions of interest in security forums [6]. A very recent work [26] uses embedding techniques to identify and classify threads of interest in a forum, which is a related, but different problem.

Some research efforts use embedding techniques to identify vulnerabilities and study the evolution of cyber-security attacks [27], [8] using security and CVE reports, and also web-blogs and databases from the darkweb.

Conclusion

We propose an iterative keyword expansion approach (IKEA) based on embedding to identify keywords relevant to

TABLE VIII: Query expansion: Evaluating IKEA on the Fire dataset.

	IKEA		AQE		QM		No-Expansion	
	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG
@10	**0.362**	**0.799***	**0.362**	0.713	0.261	0.637	0.203	0.483
@20	**0.358***	0.665	0.318	**0.681**	0.223	0.611	0.188	0.337
@30	**0.346***	0.598	0.326	**0.604**	0.236	0.529	0.188	0.261
@40	**0.335**	**0.579**	0.328	0.577	0.223	0.529	0.183	0.249
@50	**0.412***	**0.571***	0.268	0.555	0.162	0.529	0.177	0.249

an initial set of keywords. Its novelty is three-fold: (a) we use two similarity expansions in the word-word and post-post spaces, (b) we use two iterative approaches for identifying similar words and posts, and (c) we provide a flexible processing that empowers the user to finetune its outcome.

We evaluate our method with real data and we show that: (a) our approach works well with a MAP above 0.82 and NDCG above 0.85 on average, (b) IKEA outperforms the state-of-art approaches in almost all cases and often with significant difference, and (c) IKEA exhibits superior performance as a component in a query expansion task using the Fire dataset.

We see our approach as an effective building block in the space of information retrieval. We intend to fully explore its capabilities in a wide range of: (a) queries for different user preferences, beyond the two we saw here, and (b) types of data and domains. For the latter, the nature of the data (e.g. legal forum or medical journals) can introduce both new challenges and opportunities in identifying the right keywords.

REFERENCES

[1] F. Diaz, B. Mitra, and N. Craswell, "Query expansion with locally-trained word embeddings," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 367–377. [Online]. Available: https://www.aclweb.org/anthology/P16-1035

[2] S. Kuzi, A. Shtok, and O. Kurland, "Query expansion using word embeddings," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '16. New York, NY, USA: ACM, 2016, pp. 1929–1932. [Online]. Available: http://doi.acm.org/10.1145/2983323.2983876

[3] D. Roy, D. Paul, M. Mitra, and U. Garain, "Using word embeddings for automatic query expansion," *ArXiv*, vol. abs/1606.07608, 2016.

[4] K. Bennani-Smires, C. Musat, A. Hossmann, M. Baeriswyl, and M. Jaggi, "Simple unsupervised keyphrase extraction using sentence embeddings," in *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 221–229. [Online]. Available: https://www.aclweb.org/anthology/K18-1022

[5] C. Florescu and C. Caragea, "PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents," in *ACL 2017*, Vancouver, Canada, Jul. 2017, pp. 1105–1115.

[6] J. Gharibshah, E. E. Papalexakis, and M. Faloutsos, "Ripex: Extracting malicious ip addresses from security forums using cross-forum learning," in *PAKDD'18*. Springer International Publishing, 2018.

[7] R. S. Portnoff, S. Afroz, G. Durrett, J. K. Kummerfeld, T. Berg-Kirkpatrick, D. McCoy, K. Levchenko, and V. Paxson, "Tools for automated analysis of cybercriminal markets," in *WWW '17*, 2017, p. 657–666.

[8] N. Tavabi, P. Goyal, M. Almukaynizi, P. Shakarian, and K. Lerman, "Darkembed: Exploit prediction with neural language models," in *Proceedings of AAAI Conference on Innovative Applications of AI (IAAI2018)*, 2018.

[9] S. Pastrana, D. R. Thomas, A. Hutchings, and R. Clayton, "Crimebb: Enabling cybercrime research on underground forums at scale," in *Proceedings of the 2018 World Wide Web Conference*, ser. WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, p. 1845–1854. [Online]. Available: https://doi.org/10.1145/3178876.3186178

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[11] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ser. ICML'14. JMLR.org, 2014, p. II–1188–II–1196.

[12] M. Pagliardini, P. Gupta, and M. Jaggi, "Unsupervised learning of sentence embeddings using compositional n-gram features," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, ser. NAACL'18. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 528–540.

[13] A. Rajaraman and J. Jure Leskovec, Jure Ullman D, *Data Mining*. Cambridge University Press, 2011, p. 1–17.

[14] W. G. Stillwell, D. A. Seaver, and W. Edwards, "A comparison of weight approximation techniques in multiattribute utility decision making," *Organizational Behavior and Human Performance*, vol. 28, no. 1, pp. 62 – 77, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0030507381900155

[15] V. Lavrenko and W. B. Croft, "Relevance based language models," in *SIGIR '01*, New York, NY, USA, 2001.

[16] M. Carvalho, F. A. Barros, and R. B. C. Prudêncio, "A process for building domain specific thesauri for query expansion to mine SW documents repositories within an industrial environment," in *SBES '21: 35th Brazilian Symposium on Software Engineering, Joinville, Santa Catarina, Brazil, 2021*, C. D. Vasconcellos, K. G. Roggia, V. Collere, and P. Bousfield, Eds. ACM, 2021, pp. 21–26. [Online]. Available: https://doi.org/10.1145/3474624.3477057

[17] X. Yin, H. Wang, P. Yin, H. Zhu, and Z. Zhang, "A co-occurrence based approach of automatic keyword expansion using mass diffusion," *Scientometrics*, vol. 124, no. 3, pp. 1885–1905, 2020. [Online]. Available: https://doi.org/10.1007/s11192-020-03601-7

[18] M. ALMasri, C. Berrut, and J.-P. Chevallet, "A comparison of deep learning based query expansion with pseudo-relevance feedback and mutual information," in *ECIR*, vol. 9626, 03 2016, pp. 709–715.

[19] G. Xun, Y. Li, W. X. Zhao, J. Gao, and A. Zhang, "A correlated topic model using word embeddings," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, ser. IJCAI'17. AAAI Press, 2017, p. 4207–4213.

[20] A. B. Dieng, F. J. R. Ruiz, and D. M. Blei, "Topic modeling in embedding spaces," *arXiv preprint arXiv:1907.04907*, 2019.

[21] Y. Luo, H. Zhao, and J. Zhan, "Named entity recognition only from word embeddings," 2019.

[22] J. Shang *et al.*, "Learning named entity tagger using domain-specific dictionary," in *EMNLP*, 2018.

[23] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Unsupervised named-entity extraction from the web: An experimental study," *Artif. Intell.*, vol. 165, no. 1, p. 91–134, Jun. 2005.

[24] D. Nadeau, P. D. Turney, and S. Matwin, "Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity," 2006, pp. 266–277. [Online]. Available: http://cogprints.org/5025/

[25] M. Motoyama, D. McCoy, K. Levchenko, S. Savage, and G. M. Voelker, "An analysis of underground forums," ser. IMC '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 71–80.

[26] J. Gharibshah, E. E. Papalexakis, and M. Faloutsos, "Rest: A thread embedding approach for identifying and classifying user-specified information in security forums," 2020.

[27] Y. Shen *et al.*, "Attack2vec: Leveraging temporal word embeddings to understand the evolution of cyberattacks," in *USENIX Security'19*, Santa Clara, CA, 2019, pp. 905–921.